

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

طراحی صفحات وب (پیشرفته)

شاخه: کاردانش

زمینه: خدمات

گروه تحصیلی: کامپیوتر

زیرگروه: کامپیوتر

رشته مهارتی: طراحی صفحات وب

شماره رشته مهارتی: ۳-۱۷-۱۰۱-۳۱۲

کد رایانه‌ای رشته مهارتی: ۶۱۳۷

نام استاندارد مهارت مبنا: طراح صفحات وب (پیشرفته)

کد استاندارد متولی: ۱۵/۴/۲-۸۷، ف، ه

عملی: ۰۴۷۹

شماره درس: نظری: ۰۴۷۸

عنوان و نام پدیدآور: طراحی صفحات وب ... [کتاب‌های درسی] کد استاندارد متولی ۱۵/۴/۲-۸۷ - ف، ه / مؤلف زهرا سلیمی زاده؛ [برای] وزارت آموزش و پرورش، سازمان پژوهش و برنامه‌ریزی آموزشی؛ برنامه‌ریزی محتوا و نظارت بر تألیف، دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش. مشخصات نشر: تهران: مؤسسه فرهنگی و هنری دیباگران تهران، ۱۳۸۹.
مشخصات ظاهری: ۲ ج.: مصور.
شابک دوره: ۵-۰۴۸-۱۲۴-۶۰۰-۹۷۸؛ ج. ۱: ۷-۰۵۷-۱۲۴-۶۰۰-۹۷۸؛ ج. ۲: ۴-۰۵۸-۱۲۴-۶۰۰-۹۷۸
وضعیت فهرست‌نویسی: فیبا
یادداشت: ج. ۲ (چاپ اول: ۱۳۸۹) (فیبا).
مندرجات: ج. ۱. مقدماتی. - ج. ۲. پیشرفته.
موضوع: وب -- سایت‌ها -- طراحی
شناسه افزوده: سلیمی زاده، زهرا، ۱۳۵۹ -
شناسه افزوده: سازمان پژوهش و برنامه‌ریزی آموزشی. دفتر برنامه‌ریزی درسی آموزش‌های فنی و حرفه‌ای و کاردانش
شناسه افزوده: سازمان پژوهش و برنامه‌ریزی آموزشی
رده‌بندی کنگره: ۴۳ ۱۳۸۹ ط/TK5۱۰۵/۸۸۸
رده‌بندی دیدی: ۳۷۳ / ک ۱۵/۴/۱/۸۷
شماره کتابشناسی: ملی: ۲۰۷۰۵۶۵

سازمان پژوهش و برنامه‌ریزی آموزشی

همکاران محترم و دانش‌آموزان عزیز:

پیشنهادها و نظرهای خود را درباره محتوای این کتاب به نشانی: تهران - صندوق پستی شماره ۱۵ / ۸۷۴ دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش، ارسال فرمایند.
پیام‌نگار (ایمیل) : tvoccd@.roshd.ir وب‌گاه (وب‌سایت) : WWW.tvoccd.medu.ir

برنامه‌ریزی محتوا و نظارت بر تألیف: دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش

عنوان و شماره کتاب: طراحی صفحات وب (پیشرفته) - ۶۱۲/۱۰

مجری: مؤسسه فرهنگی هنری دیباگران تهران

شماره درس: ۰۴۷۸ ، ۰۴۷۹

مؤلف: مهندس زهرا سلیمی‌زاده

ویراستار ادبی: فاطمه پورعبدل

صفحه‌آرا: آمنه درویش

طراح جلد: بیتا اشرفی مقدم

محتوای این کتاب در کمیسیون جلسه مورخ ۸۹/۳/۲۳ کمیسیون تخصصی رشته کامپیوتر دفتر

برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش با عضویت: بتول عطاران، محمدرضا

شکرریز، محمدعباسی، الهه اسماعیل‌زاده، نادیا ماجدی و اعظم امیدوار تأیید شده است.

نوبت و سال چاپ:

چاپخانه:

تبراز:

ناشر: مؤسسه فرهنگی هنری دیباگران تهران

نشانی ناشر: تهران - سعادت‌آباد - میدان کاج - خیابان سرو شرقی - روبه‌روی خیابان علامه - پلاک ۴۹

آدرس الکترونیکی: www.dibagaran.mft.info

تلفن: ۲۲۰۹۸۴۶۷-۷ دورنگار: ۲۲۰۹۸۴۴۸ صندوق پستی: ۱۴۶۵۵/۴۶۶

نظارت بر چاپ و توزیع: اداره کل چاپ و توزیع کتاب‌های درسی، سازمان پژوهش و برنامه‌ریزی آموزشی

نشانی: تهران - خیابان ایرانشهر شمالی - ساختمان شماره ۴ آموزش و پرورش (شهید موسوی)

تلفن: ۸۸۳۱۱۶۱-۹ دورنگار: ۸۸۳۰۹۲۶۶ صندوق پستی: ۱۵۸۴۷۴۷۳۵۹

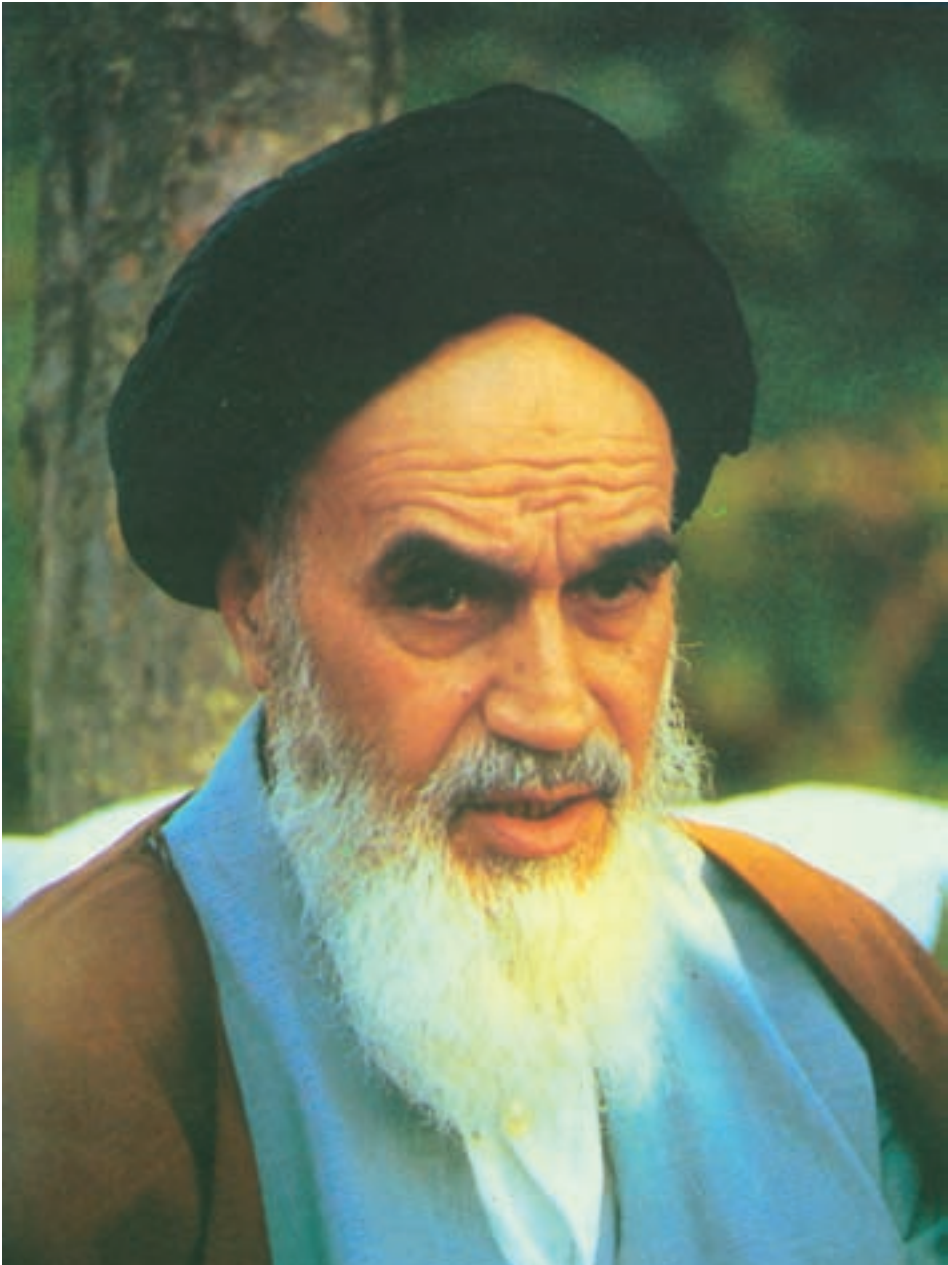
آدرس الکترونیکی: www.chap.roshd.ir

شابک: ۹۷۸ - ۶۰۰ - ۱۲۴ - ۰۵۸ - ۴

شابک دوره: ۹۷۸ - ۶۰۰ - ۱۲۴ - ۰۴۸ - ۵

نشانی دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش صندوق پستی شماره: ۱۵۸۷۵/۴۸۷۴ کلیه حقوق مربوط به تألیف، نشر و چاپ این اثر متعلق به ناشر و سازمان پژوهش و برنامه‌ریزی آموزشی است.

(حق طبع محفوظ است)



بدانید مادام که در احتیاجات صنایع پیشرفته، دست خود را پیش دیگران دراز کنید و به در یوزگی عمر را بگذارید، قدرت ابتکار و پیشرفت در اختراعات در شما شکوفا نخواهد شد.

امام خمینی «قدس سرّه الشّریف»

مجموعه کتاب‌های درسی رشته کامپیوتر شاخه کاردانش

(استاندارد وزارت فرهنگ و ارشاد اسلامی)

رشته تصویرسازی	رشته طراحی صفحات وب	رشته تولید چندرسانه‌ای
مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات
سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی
اطلاعات و ارتباطات	اطلاعات و ارتباطات	اطلاعات و ارتباطات
واژه‌پرداز Word 2007	واژه‌پرداز Word 2007	واژه‌پرداز Word 2007
صفحه گسترده Excel 2007	صفحه گسترده Excel 2007	صفحه گس‌ترده Excel 2007
ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007
نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی
بانک اطلاعاتی Access 2007	بانک اطلاعاتی Access 2007	بانک اطلاعاتی Access 2007
سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته
برنامه‌نویسی مقدماتی	برنامه‌نویسی مقدماتی	برنامه‌نویسی مقدماتی
طراحی امور گرافیکی با رایانه	طراحی امور گرافیکی با رایانه	طراحی امور گرافیکی با رایانه
کاربر FreeHand	کاربر Flash	کاربر Flash
کاربر CorelDraw	طراحی صفحات وب مقدماتی	کاربر Director
	طراحی صفحات وب پیشرفته	میکس رایانه‌ای

مجموعه کتاب‌های درسی رشته کامپیوتر شاخه کاردانش

(استاندارد وزارت کار و امور اجتماعی)

رشته تصویرسازی	رشته طراحی صفحات وب	رشته تولید چندرسانه‌ای	رشته برنامه‌نویسی پایگاه داده
مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات	مفاهیم پایه فناوری اطلاعات
سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی	سیستم‌عامل مقدماتی
اطلاعات و ارتباطات	اطلاعات و ارتباطات	اطلاعات و ارتباطات	اطلاعات و ارتباطات
سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته	سیستم‌عامل پیشرفته
واژه‌پرداز Word 2007	واژه‌پرداز Word 2007	واژه‌پرداز Word 2007	واژه‌پرداز Word 2007
صفحه گسترده Excel 2007	صفحه گسترده Excel 2007	صفحه گسترده Excel 2007	صفحه گسترده Excel 2007
ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007	ارایه مطالب PowerPoint 2007
نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی	نرم‌افزارهای اداری تکمیلی
طراح گرافیک رایانه‌ای	طراح گرافیک رایانه‌ای	طراح گرافیک رایانه‌ای	برنامه‌نویسی مقدماتی
شهروند الکترونیکی	نرم‌افزار گرافیکی Flash Mx	شهروند الکترونیکی	برنامه‌نویسی ویژوال بیسیک پیشرفته (جلد اول)
نرم‌افزار گرافیکی FreeHand	طراحی مقدماتی صفحات وب	نرم‌افزار گرافیکی Director	برنامه‌نویسی ویژوال بیسیک پیشرفته (جلد دوم)
نرم‌افزار گرافیکی CorelDraw	رایانه کار Interdev	تدوین فیلم و صدا SSP	مدیریت پایگاه داده
نرم‌افزار گرافیکی Flash Mx	رایانه کار Dreamweaver	نرم‌افزار گرافیکی Flash Mx	مهارت عمومی برنامه‌نویسی
	رایانه کار CIW	نرم‌افزار گرافیکی Authorware	

فهرست مطالب

مقدمه ناشر

مقدمه

واحد کار اول: نصب و راه اندازی MySQL و کار با آن

۲	۱-۱ آشنایی با MySQL
۴	۱-۲ نصب MySQL
۱۲	واژه نامه
۱۲	خلاصه مطالب
۱۳	آزمون نظری
۱۳	آزمون عملی

واحد کار دوم: پایگاه داده

۱۶	مقدمه
۱۶	۲-۱ مدل های پایگاه داده
۲۳	۲-۲ انواع ماهیت ارتباط
۲۴	۲-۳ محیط عملیاتی تولیدکنندگان قطعات صنعتی
۲۷	۲-۴ نرمال سازی بانک های اطلاعاتی
۳۴	واژه نامه
۳۴	خلاصه مطالب
۳۶	آزمون نظری

واحد کار سوم: آشنایی با دستورات کار با پایگاه داده

۳۸	مقدمه
۳۸	۳-۱ آشنایی با SQL
۴۰	۳-۲ دسترسی به MySQL از طریق خط فرمان
۴۱	۳-۳ دستور create
۴۵	۳-۴ دستور select
۴۷	۳-۵ دستور where
۴۹	۳-۶ دستور insert
۵۱	۳-۷ دستور update

۵۳	۳-۸ دستور delete
۵۴	۳-۹ دستور order by
۵۶	۳-۱۰ دستور and و or
۵۸	۳-۱۱ دستور in
۵۸	۳-۱۲ دستور between ... and
۶۰	۳-۱۳ alias ویژگی
۶۱	۳-۱۴ پیوند جدول‌ها
۶۷	۳-۱۵ دستور join
۶۸	۳-۱۶ Select های تودرتو
۷۰	۳-۱۷ دستور union
۷۲	۳-۱۸ دستور drop
۷۳	۳-۱۹ دستور alter
۷۴	۳-۲۰ توابع درون ساخته SQL
۸۵	واژه‌نامه
۸۵	خلاصه مطالب
۸۶	آزمون نظری
۸۸	آزمون عملی

واحد کار چهارم: نصب APACHE

۹۰	مقدمه
۹۰	۴-۱ نصب Apache
۹۶	واژه‌نامه
۹۶	خلاصه مطالب
۹۷	آزمون نظری
۹۸	آزمون عملی

واحد کار پنجم: نصب PHP

۱۰۰	مقدمه
۱۰۱	۵-۱ آشنایی با زبان PHP
۱۰۷	۵-۲ نصب PHP در لینوکس
۱۰۸	۵-۳ فایل php.ini در ویندوز
۱۰۹	۵-۴ نصب PHP بدون استفاده از سرویس دهنده‌ها
۱۱۲	۵-۵ پیکربندی Apache در ویندوز
۱۱۲	۵-۶ پیکربندی Apache در لینوکس
۱۱۲	۵-۷ بررسی صحت عملکرد PHP
۱۱۴	واژه‌نامه
۱۱۴	خلاصه مطالب
۱۱۵	آزمون نظری

واحد کار ششم: کار با متغیرها، عملگرها و ساختارهای کنترلی

۱۱۸	مقدمه
۱۱۸	۶-۱ شروع برنامه‌نویسی در PHP
۱۲۲	۶-۲ متغیرها
۱۲۷	۶-۳ عملگرها
۱۳۰	۶-۴ کار با ساختارهای کنترلی
۱۳۵	۶-۵ حلقه‌ها
۱۳۹	۶-۶ دستورات break و continue
۱۴۰	۶-۷ متغیرهای سراسری
۱۴۱	واژه‌نامه
۱۴۱	خلاصه مطالب
۱۴۳	آزمون نظری
۱۴۴	آزمون عملی

واحد کار هفتم: برقراری ارتباط با کاربر

۱۴۶	مقدمه
۱۴۶	۷-۱ دریافت اطلاعات از فرم
۱۴۹	۷-۲ اعتبارسنجی داده‌های ورودی
۱۵۰	۷-۳ ارسال نامه الکترونیکی از طریق PHP
۱۵۳	۷-۴ بارگذاری (Upload) فایل‌ها از طریق فرم
۱۶۰	۷-۵ طراحی فرم بازخورد (Feedback)
۱۶۲	واژه‌نامه
۱۶۲	خلاصه مطالب
۱۶۳	آزمون نظری
۱۶۳	آزمون عملی

واحد کار هشتم: مدیریت فایل‌ها و پوشه‌ها

۱۶۶	مقدمه
۱۶۶	۸-۱ کار با فایل‌ها
۱۶۸	۸-۲ کار با پوشه‌ها
۱۷۰	واژه‌نامه
۱۷۰	خلاصه مطالب
۱۷۱	آزمون نظری
۱۷۱	آزمون عملی

واحد کار نهم: کار با توابع

۱۷۴	مقدمه
۱۷۴	۹-۱ توابع رشته‌ای
۱۷۸	۹-۲ توابع تاریخ و زمان
۱۸۱	۹-۳ توابع عددی
۱۸۴	۹-۴ تعریف توابع جدید

۱۸۸	واژه‌نامه
۱۸۸	خلاصه مطالب
۱۸۹	آزمون نظری
۱۹۰	آزمون عملی

واحد کار دهم: کار با پایگاه داده

۱۹۲	مقدمه
۱۹۲	۱۰-۱ مجوزهای کاربری
۱۹۶	۱۰-۲ لیست پایگاه‌های داده
۱۹۸	۱۰-۳ مدیریت پایگاه داده
۱۹۹	۱۰-۴ ساخت جدول phpMyAdmin
۲۰۱	۱۰-۵ ارتباط با پایگاه داده در Dreamweaver
۲۰۶	۱۰-۶ مشاهده اطلاعات پایگاه داده
۲۰۸	۱۰-۷ درج اطلاعات در پایگاه داده
۲۰۹	۱۰-۸ حذف رکورد از پایگاه داده
۲۱۳	۱۰-۹ ویرایش اطلاعات پایگاه داده
۲۱۶	مطالعه آزاد
۲۱۶	۱۰-۱۰ ایجاد ارتباط با سیستم مدیریت پایگاه داده MySQL
۲۱۸	۱۰-۱۱ ایجاد یک پایگاه داده و اجزای آن
۲۲۵	۱۰-۱۲ انتخاب و نمایش اطلاعات جدول
۲۲۸	۱۰-۱۳ به روزرسانی اطلاعات جدول
۲۲۹	۱۰-۱۴ حذف رکورد
۲۳۱	واژه‌نامه
۲۳۱	خلاصه مطالب
۲۳۲	آزمون نظری
۲۳۳	آزمون عملی

واحد کار یازدهم: به‌کارگیری کوکی‌ها و Session‌ها

۲۳۶	مقدمه
۲۳۷	۱-۱۱ ایجاد کوکی
۲۴۱	۲-۱۱ جلسه Session
۲۴۴	واژه‌نامه
۲۴۴	خلاصه مطالب
۲۴۵	آزمون نظری
۲۴۶	آزمون عملی

واحد کار دوازدهم: ارزیابی وب‌سایت و صفحات وب

۲۴۸	مقدمه
۲۴۸	۱-۱۲ ارزیابی صفحات وب
۲۵۳	واژه‌نامه
۲۵۳	خلاصه مطالب
۲۵۴	آزمون نظری
۲۵۴	آزمون عملی
۲۵۵	پیوست
۲۹۰	پاسخنامه
۲۹۲	فهرست منابع

مقدمه ناشر

سپاس بیکران پروردگار را که به انسان قدرت اندیشیدن بخشید تا به یاری این موهبت راه ترقی و تعالی را ببیند و امید به این که عنایات الهی شامل حال ما باشد تا با بضاعت اندک علمی خود در خدمت جوانان و آینده‌سازان کشور عزیزمان باشیم.

یکی از بارزترین ویژگی‌های عصر حاضر، حضور گسترده کامپیوتر در کلیه عرصه‌های فعالیت انسان است به گونه‌ای که انجام برخی از کارها، بدون استفاده از کامپیوتر قابل تصور نیست. کامپیوتر به عنوان ابزاری قدرتمند، سرعت و دقت کارها را فوق‌العاده افزایش داده و گذرگاه‌های صعب‌العبور علم را به شاهراه‌های هموار مبدل ساخته است. به همین دلیل در جهان کنونی، آموزش و یادگیری علوم کامپیوتر یک ضرورت اجتناب‌ناپذیر است.

در همین راستا دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش بر اساس موافقت‌نامه‌ای، تألیف کتاب‌های مهارت‌های رایانه کار پیشرفته و برنامه‌نویسی و پژوهش‌های مقدماتی شاخه کاردانش را به مؤسسه فرهنگی هنری دیباگران تهران به‌عنوان یکی از ناشران برگزیده محول کرده که افتخاری بزرگ است. کتاب حاضر با همکاری جمعی از اساتید، متخصصان و مهندسان مجرب رشته کامپیوتر تألیف و محتوای آن در کمیسیون تخصصی برنامه‌ریزی و تألیف کتاب‌های درسی رشته کامپیوتر دفتر برنامه‌ریزی و تألیف آموزش‌های فنی و حرفه‌ای و کاردانش مورد بررسی و تصویب قرار گرفته است. طراحی کتاب‌ها براساس ساختار آموزش‌های پیمان‌های (مادولار) انجام گرفته و ساختار آن بر اساس توانایی‌های مورد انتظار در استانداردهای مهارتی طراحی شده است. این کتاب‌ها حتی‌المقدور به صورت خودآموز و خودمحتوا سازماندهی شده و تلاش بر این است که کتاب‌های آموزش گام به گام، به همراه مثال‌ها، تمرین‌های عملی و کاربردی برای کارهای آزمایشگاهی و کارگاهی به انضمام سؤالات پیش‌آزمون و آزمون پایانی، مجموعه منسجمی از هر پیمان‌ها را ارائه دهد به طوری که دانش‌آموزان پس از پایان هر پیمان‌ها، از مهارت کافی برای کار با موضوع پیمان‌ها برخوردار باشند.

در خاتمه ضمن سپاسگزاری از شما دانش‌پژوه گرامی درخواست می‌نماید با مراجعه به آدرس dibagaran.mft.info (ارتباط با مشتری) فرم نظرسنجی را برای کتابی که در دست دارید تکمیل و ارسال نموده، انتشارات دیباگران تهران را که جلب رضایت و وفاداری مشتریان را هدف خود می‌داند، یاری فرمایید.

امیدواریم همواره بهتر از گذشته خدمات و محصولات خود را تقدیم حضورتان نماییم.

مدیر انتشارات

مؤسسه فرهنگی هنری دیباگران تهران

publishing@mftmail.com

مقدمه

امروزه با پیشرفت علم و فناوری، افزایش سطح دانش و مهارت‌های ارتباطی به عنوان دو جزء لاینفک از یکدیگر قلمداد می‌شوند، در راستای ایجاد ارتباط در عصر اطلاعات، اینترنت مناسب‌ترین و در عین حال کم‌هزینه‌ترین راه ایجاد ارتباط تعاملی به شمار می‌رود، وب به عنوان اصلی‌ترین رابط اینترنت به منظور تبادل اطلاعات در دنیای مجازی به کار می‌رود.

لذا برای بهره‌گیری مؤثر از اینترنت و ارتباط با آن بهتر است خود را به علم طراحی صفحات وب و وبسایت مجهز نماییم.

در این کتاب سعی شده است مراحل طراحی یک وبسایت به‌طور خودآموز و با بیان بسیار ساده آموزش داده شود، اما مسلماً از اشتباهات احتمالی مبرا نخواهد بود، لذا از خوانندگان گرامی این کتاب خواهشمندم اینجانب را از نظرات ارزشمند خود بهره‌مند نمایند تا در چاپ بعدی کتاب نسبت به اصلاح آن اقدام شود.

مؤلف

نصب و راه‌اندازی MySQL و کار با آن

واحد کار اول

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

- ۱ - با MySQL آشنا باشد و نگارش‌های مختلف آن را بشناسد.
- ۲ - بتواند MySQL را با نرم‌افزارهای مشابه مقایسه کند.
- ۳ - امکانات لازم برای نصب و استفاده از MySQL را بشناسد.
- ۴ - با اصول نصب MySQL در ویندوز آشنا باشد.
- ۵ - با اصول نصب MySQL در لینوکس آشنا باشد.
- ۶ - ساختار پوشه‌های MySQL را بشناسد.
- ۷ - فایل‌های ذخیره‌سازی MySQL را بشناسد.
- ۸ - پوشه داده‌ها در MySQL را بشناسد.

زمان (ساعت)	
عملی	نظری
۸	۴

۱ - ۱ آشنایی با MySQL

MySQL یکی از سیستم‌های مدیریت پایگاه داده متن باز^۱ است که برای ایجاد و مدیریت بانک‌های اطلاعاتی در سطح وسیعی استفاده می‌شود. پایگاه داده، شامل مجموعه‌ای از داده‌هاست که به صورت ساخت یافته و در قالب جداول در کنار یکدیگر قرار گرفته‌اند، در پایگاه داده برای عملیاتی مانند حذف، اضافه و ویرایش داده‌ها باید از یک سیستم مدیریت پایگاه داده‌ها مانند MySQL استفاده کرد. اگر حجم اطلاعات و داده‌ها در پایگاه داده زیاد باشد، عملیات پردازشی و محاسباتی نقش زیادی در انتخاب پایگاه داده دارند، برخی از پایگاه‌های داده مانند Access نسبت به نمونه‌های مشابه سرعت کمی در عملیات پردازشی و محاسباتی دارد، لذا به کار بردن آن در بانک‌های اطلاعاتی با حجم بالا توصیه نمی‌شود، اما بانک اطلاعاتی Oracle به علت سرعت بسیار بالا غالباً در بانک‌های اطلاعاتی بسیار بزرگ استفاده می‌شود، بانک اطلاعاتی MySQL نیز در بانک‌های اطلاعاتی نسبتاً بزرگ و تا حجم چند میلیون داده عملکرد مناسبی ارائه می‌دهد، به همین دلیل در بین طراحان وب از محبوبیت بالایی برخوردار بوده و به دلیل راحت بودن محیط طراحی بانک اطلاعاتی و مدیریت آن همچنان مورد توجه طراحان بانک اطلاعاتی است.

۱-۱-۱ ویژگی‌های اصلی MySQL

- یکی از ویژگی‌های اصلی MySQL این است که به صورت متن باز بوده و آن را می‌توان به طور رایگان از اینترنت دریافت کرد.
- این سیستم مدیریت پایگاه داده می‌تواند در محیط‌های سیستم‌های عامل مختلف از جمله ویندوز و لینوکس به خوبی کار کند و به دلیل وجود API‌های (Application Programming Interface) متعدد، می‌تواند توسط زبان‌های برنامه‌نویسی مختلف از جمله Perl، PHP، java، C++ و C استفاده شود.
- MySQL می‌تواند در سیستم‌هایی که چند پردازنده دارند استفاده شود و حتی می‌تواند از چندین CPU برای اجرا استفاده کند.

- به دلیل استفاده از تکنیک‌های سریع دسترسی به حافظه می‌تواند با سرعت بالایی اطلاعات مورد نیاز کاربران را در اختیار آن‌ها قرار دهد.
- از امنیت نسبتاً بالایی برخوردار بوده و می‌تواند تا حد زیادی مانع دسترسی هکرها به اطلاعات درون بانک اطلاعاتی شود.
- نسخه‌های قبل از MySQL 4.1.3 از زبان فارسی پشتیبانی نمی‌کردند و برای ذخیره، نمایش و مرتب‌سازی داده‌های فارسی از قالب مخصوص زبان عربی استفاده می‌شد، به همین دلیل برای نمایش داده‌های فارسی که دارای حروف "گ، چ، پ، ژ" بودند مشکل پیش می‌آمد اما در نسخه‌های پس از آن، امکانات فارسی با استاندارد UTF8 به MySQL وارد شده و مشکلات زبان فارسی نیز حل شد

۲-۱-۱ نسخه‌های MySQL

- MySQL دارای نسخه‌های متعددی است که تفاوت‌هایی نیز با یکدیگر دارند، به‌طور کلی گروهی از نسخه‌ها فقط برای توسعه استفاده می‌شوند و گروهی دیگر می‌توانند به عنوان یک سیستم مدیریت پایگاه داده دائمی برای طراحی سایت به کار روند.
- اگر به عنوان نسخه‌های MySQL دقت کنید می‌بینید که در نامگذاری آن‌ها یک الگوی مشخص رعایت شده است، به عنوان مثال MySQL 5.1.25.
- عدد اول (5):** نسخه اصلی و توصیف‌کننده قالب فایل است. این به آن معناست که تمام نسخه‌های MySQL... دارای یک توصیف‌کننده قالب فایل یکسان هستند و الزاماً اینگونه نیست که MySQL4... قدیمی‌تر از MySQL5... باشد.
- عدد دوم (1):** میزان انتشار را معین می‌کند، شماره اول و دوم با همدیگر شماره سری انتشار را تشکیل می‌دهند.
- عدد سوم (25):** شماره ویرایش در سری انتشار است. یعنی هر بار که ویرایشی روی هر نسخه MySQL صورت می‌گیرد، به این شماره یکی افزوده می‌شود.
- پس از گروه اعداد فوق از یک پسوند هم در نامگذاری نسخه‌های MySQL استفاده می‌شود که عبارتند از:
- alpha:** نسخه‌ای که از این نوع باشد، نشان‌دهنده این است که به آن کدهای جدیدی افزوده شده است اما هنوز به‌طور کامل عیب‌یابی نشده‌اند و هنوز در دوران تست و عیب‌یابی به سر می‌برند.

beta: از لحاظ خصوصیات و ویژگی‌ها با نسخه alpha از همان نرم‌افزار تفاوت ندارد و فقط دوران تست را پشت سر گذاشته و به عنوان نسخه نهایی عرضه شده است.
gama: همان نسخه alpha است که به منظور صحت عملکرد بررسی و تعمیر شده است.

۲-۱ نصب MySQL

برای نصب و راه‌اندازی MySQL نیاز به وجود قطعات سخت‌افزاری پیشرفته نیست و این سیستم مدیریت پایگاه داده می‌تواند روی تمام سخت‌افزارهای امروزی نصب شده و به راحتی مورد استفاده قرار بگیرد.

۱-۲-۱ نصب در لینوکس

برای نصب MySQL در لینوکس بهتر است از بسته‌های RPM استفاده کنید، این بسته‌ها برای پایگاه‌های داده طراحی شده‌اند و فقط روی برخی از نسخه‌های لینوکس قابل اجرا هستند. در حال حاضر برای برخی نسخه‌های لینوکس از جمله Suse7.3 بهتر کار می‌کند. هنگام نصب MySQL در لینوکس باید یک گروه کاری و یک کاربر جدید برای کار با MySQL ایجاد کنید.

۲-۲-۲ نصب در ویندوز

اگر قبلاً روی سیستم خود MySQL را نصب نکرده‌اید مراحل زیر را برای نصب آن انجام دهید:

در مسیر فایل‌های نصب MySQL روی آیکن مربوط به نصب دو بار کلیک کنید، پس از مدت کوتاهی کادر محاوره شکل ۱-۱ ظاهر می‌شود.
روی دکمه Next کلیک کنید.

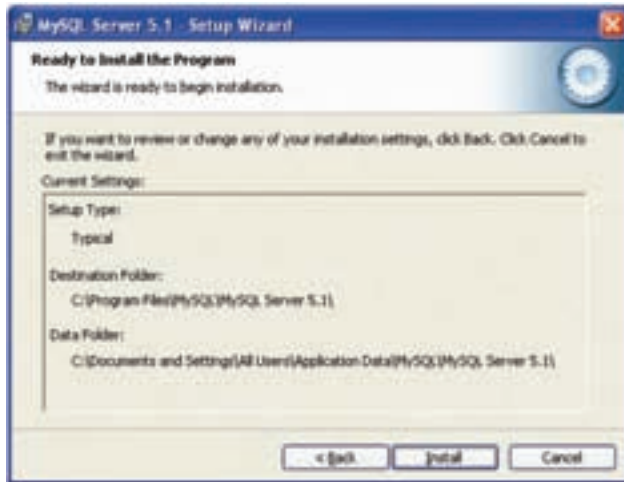


شکل ۱-۱



شکل ۱-۲

دکمه رادیویی Typical را انتخاب کرده، روی دکمه Next کلیک کنید.



شکل ۱-۳

در کادرمحاوره شکل ۱-۳ روی دکمه Install کلیک کنید.
در کادرمحاوره شکل ۱-۴ روی دکمه Next کلیک کنید.
در کادرمحاوره شکل ۱-۵ دقت کنید که حتماً کادر علامت
Configure the MySQL Server now انتخاب شده باشد.
روی دکمه Finish کلیک کنید.



شکل ۱-۴



شکل ۱-۵



شکل ۱-۶

در شکل ۱-۶ روی دکمه Next کلیک کنید.
دقت کنید که دکمه رادیویی Detailed Configuration انتخاب شده باشد، سپس روی دکمه Next کلیک کنید.

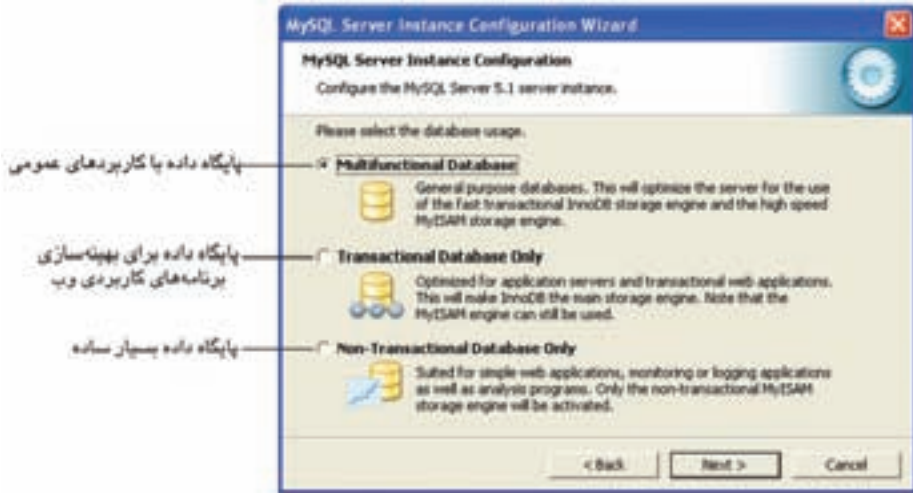


شکل ۱-۷

پس از انتخاب دکمه رادیویی Developer Machine در شکل ۱-۸، روی دکمه Next کلیک کنید.
در کادر محاوره شکل ۱-۹ دکمه رادیویی Multifunctional Database را انتخاب کرده، روی دکمه Next کلیک کنید.



شکل ۱-۸

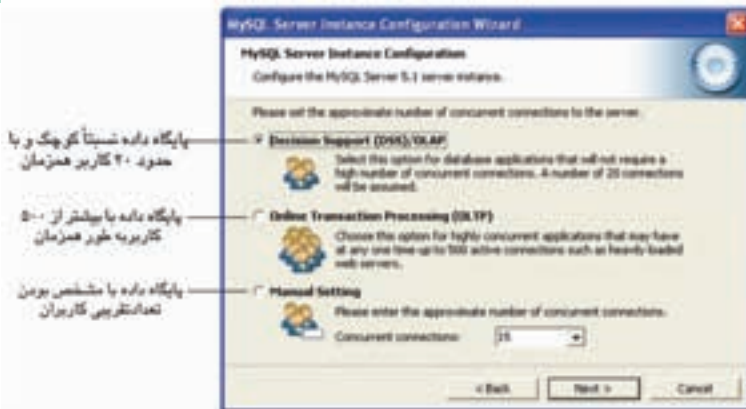


شکل ۹-۱



شکل ۱۰-۱

در کادر محاوره شکل ۱۰-۱ روی دکمه Next کلیک کنید.
 دقت کنید دکمه رادیویی Decision Support (DSS)/OLAP در شکل ۱۱-۱ انتخاب شده باشد، سپس روی دکمه Next کلیک کنید.
 در شکل ۱۲-۱ در حالتی که هر دو کادر علامت انتخاب شده باشند، روی دکمه Next کلیک کنید.



شکل ۱-۱۱



شکل ۱-۱۲



شکل ۱-۱۳

در شکل ۱-۱۳ برای پشتیبانی از حداکثر زبان‌ها، دکمه رادیویی Best Support For Multilingualism را انتخاب کرده، سپس روی دکمه Next کلیک کنید.



شکل ۱-۱۴

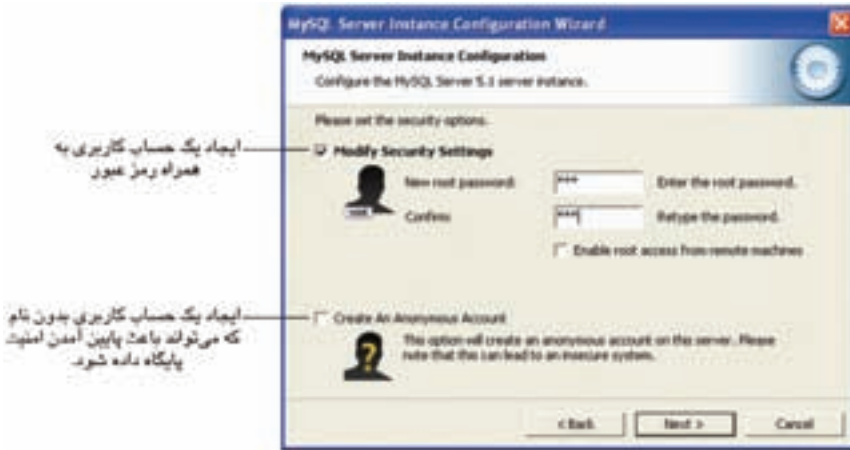
در شکل ۱-۱۴ هر سه کادر علامت موجود را انتخاب کرده، سپس روی دکمه Next کلیک کنید.

در شکل ۱-۱۵ دقت کنید کادر علامت Modify Security Settings در حالت انتخاب باشد، سپس در هر دو کادر متنی زیر آن، کلمه رمز را وارد کنید.



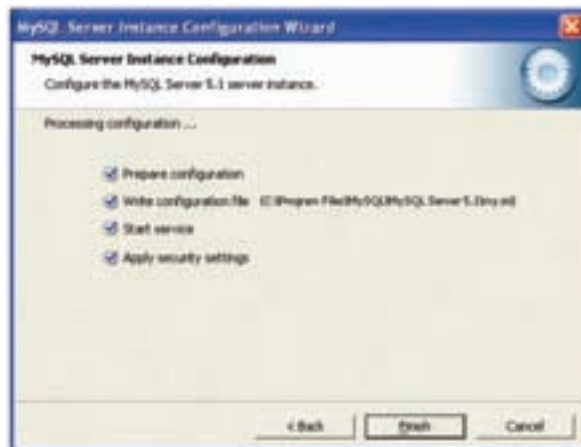
شکل ۱-۱۵

روی دکمه Next کلیک کنید.



شکل ۱-۱۶

در کادرمحاوره شکل ۱-۱۶ پس از تعیین رمز عبور، روی دکمه Next کلیک کنید.



شکل ۱-۱۷

و در نهایت در کادرمحاوره شکل ۱-۱۷ روی دکمه Finish کلیک کنید.

واژه‌نامه

Configure	پیکربندی
Details	جزئیات
Developer	توسعه‌دهنده
Interface	رابط کاربری
Multifunctional	چندکاره
Typical	نمونه

خلاصه مطالب

- MySQL یکی از سیستم‌های مدیریت پایگاه داده متن باز است که برای ایجاد و مدیریت بانک‌های اطلاعاتی استفاده می‌شود.
- در انتخاب پایگاه داده، حجم اطلاعات و داده‌ها مستقیماً دخالت دارد.
- عنوان نسخه‌های MySQL براساس یک الگوی مشخص تعیین می‌شود.
- عدد اول، نسخه اصلی و توصیف‌کننده قالب فایل است.
- عدد دوم، سطح انتشار را تعیین می‌کند.
- عدد سوم، شماره ویرایش در سری انتشار است.

آزمون نظری

- ۱ - کدام یک از گزینه‌های زیر در انتخاب پایگاه داده مؤثرتر است؟
- الف- متن باز بودن
ب- داشتن امکان نصب آسان
ج- وجود API
د- عملیات پردازشی و محاسباتی
- ۲ - MySQL به چه دلیل توسط زبان‌های برنامه‌نویسی می‌تواند مورد استفاده قرار گیرد؟
- الف- متن باز بودن
ب- وجود APIهای متعدد
ج- عملیات پردازشی و محاسباتی
د- امکان نصب سریع
- ۳ - در MySQL 4.1.5، عدد ۴ نشان‌دهنده چیست؟
- الف- سطح انتشار
ب- شماره ویرایش
ج- زمان انتشار
د- توصیف‌کننده قالب فایل
- ۴ - روی کدام یک از انواع نسخه‌های زیر از MySQL علاوه بر تست، تعمیر نیز صورت گرفته است؟
- الف - alpha ب - beta ج - gama د - 5

آزمون عملی

- ۱ - MySQL را روی سیستم خود نصب و راه‌اندازی کنید.

پایگاه داده

واحد کار دوم

- پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:
- ۱ - انواع مدل‌های پایگاه داده را بشناسد.
 - ۲ - مفهوم پایگاه‌های داده رابطه‌ای را بداند.
 - ۳ - انواع رابطه‌ها را بشناسد.
 - ۴ - اصول طراحی پایگاه داده را بداند.
 - ۵ - با مفهوم نرمال‌سازی آشنا باشد.
 - ۶ - انواع مدل‌های نرمال اول، دوم و سوم را بشناسد.

زمان (ساعت)	
عملی	نظری
۵	۸

مقدمه

با مفاهیم کلی بانک اطلاعاتی، موجودیت، رکورد، فیلد و ... در درس بانک اطلاعاتی سال گذشته آشنا شده‌اید.


در این درس ضمن یادآوری مباحث گذشته، با مفاهیم پیشرفته‌تر بانک اطلاعاتی نیز آشنا خواهید شد. در حقیقت سیستم بانک اطلاعاتی یا پایگاه داده‌ها، سیستم کامپیوتری نگهداری اطلاعات است. پایگاه داده را می‌توان به یک قفسه الکترونیکی تشبیه کرد که اطلاعات را به‌طور منظم درون خود نگهداری می‌کند.

کاربران سیستم بانک اطلاعاتی می‌توانند اعمال مختلفی روی آن انجام دهند که نمونه‌هایی از این اعمال عبارتند از:

- افزودن فایل‌های جدید و خالی به پایگاه داده
- افزودن داده درون فایل‌هایی که از قبل ایجاد شده‌اند.
- بازیابی داده‌ها از فایل‌های موجود
- تغییر در داده‌های فایل‌های موجود
- حذف داده‌ها از فایل‌های موجود
- حذف فایل‌های پایگاه داده

۱-۲ مدل‌های پایگاه داده

مدل‌های پایگاه داده‌ها یک طرح و الگوی کاری برای کاربران پایگاه داده‌ها را در سطح منطقی مشخص می‌کنند و معمولاً سیستم‌های مدیریت پایگاه داده براساس مدل‌هایی که استفاده می‌کنند تقسیم‌بندی می‌شوند.

نکته  به برنامه کامپیوتری که برای مدیریت داده‌ها و پاسخگویی به کاربران از طریق پایگاه داده به‌کار برده می‌شود سیستم مدیریت پایگاه داده (DBMS) گفته می‌شود.

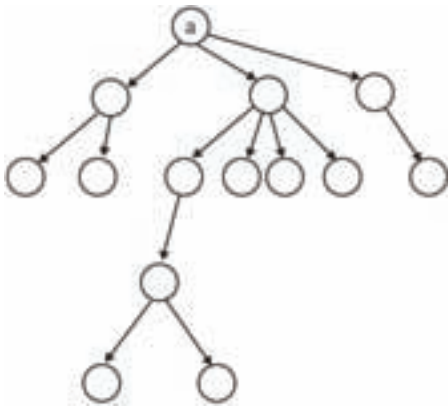
مهم‌ترین انواع مدل‌های پایگاه داده عبارتند از:

- پایگاه داده سلسله مراتبی
- پایگاه داده رابطه‌ای

- پایگاه داده شبکه‌ای
- پایگاه داده شیء‌گرا

۱-۱-۲ مدل سلسله مراتبی

این ساختار یکی از قدیمی‌ترین مدل‌های طراحی بانک اطلاعاتی است. در این مدل

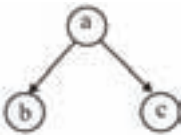


شکل ۲-۱

داده‌ها و ارتباط بین آن‌ها توسط اشاره‌گرهایی با یکدیگر مرتبط می‌شوند. شکل ۲-۱ یک نمونه مدل سلسله مراتبی است. مدل سلسله مراتبی که مشابه ساختار درخت پیاده می‌شود، دارای یک گره ریشه a و N گره فرزند است.

ارتباط بین گره‌ها از بالا به پایین صورت می‌گیرد و مسیری از گره‌های پایین‌تر به گره‌های بالاتر وجود ندارد.

هر گره پدری می‌تواند چند گره فرزند داشته باشد، اما هر گره فرزند، تنها یک گره پدر دارد.

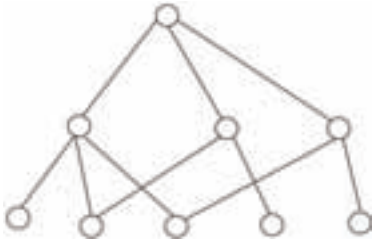


شکل ۲-۲

در شکل ۲-۲، a گره پدر و b و c گره‌های فرزند هستند. در عمل پیاده‌سازی ساختار سلسله مراتبی بسیار مشکل است، به همین دلیل پس از مدتی جای خود را به مدل‌های دیگر پایگاه داده داد و عملاً منسوخ شده است.

۱-۲-۲ پایگاه داده شبکه‌ای

این مدل در سال ۱۹۶۹ و براساس دو مفهوم مهم مجموعه‌ها و رکوردها ارائه شد. پیاده‌سازی این مدل توسط گراف انجام می‌شود.



شکل ۲-۳

مزیت این روش این است که مدل‌های ارتباطی طبیعی بیشتری را بین موجودیت‌ها فراهم می‌کند و در عین حال پیچیدگی کمتری خواهد داشت.

اگر بخواهیم این مدل را توسط ساختار درختی پیاده کنیم، می‌توانیم بگوییم که هر گره فرزند می‌تواند بیش از یک گره پدر داشته باشد.

این مدل پیچیده‌تر از مدل سلسله مراتبی است و در نتیجه عملیات ذخیره و بازیابی پیچیده‌تری نیز دارد.

۲-۱-۳ مدل رابطه‌ای

پایگاه داده رابطه‌ای از دید کاربران شامل یک مجموعه جدول است که هر جدول دارای سطرها و ستون‌هایی است. اطلاعات مربوط به هر ستون، فیلدهای جدول و اطلاعات هر سطر مربوط به یکی از رکوردهای جدول است.

جدول ۲-۱

نام خانوادگی	شماره دانش‌آموزی	نام
اقبالی	۱۳۴۵۷۹۳۲	علی
احمدی	۳۲۴۳۳۵۹۲	مریم
بهرامی	۲۴۱۲۳۷۱۴	میثم
فدایی	۱۲۳۹۲۳۴۳	زهرا

در جدول ۲-۱، سه فیلد با عناوین نام، نام خانوادگی و شماره دانش‌آموزی وجود دارد و همچنین هر سطر اطلاعات مربوط به یکی از رکوردهای جدول را نمایش داده است.

فیلد: کوچک‌ترین واحد داده ذخیره شده در بانک اطلاعاتی است. جدول ۲-۱ شامل سه فیلد ایجاد شده با عناوین «نام»، «نام خانوادگی» و «شماره دانش‌آموزی» است.

رکورد: مجموعه‌ای از فیلدهای ذخیره شده مرتبط به هم است. به عنوان مثال در جدول ۲-۱ نام خانوادگی فدایی مربوط به فردی با نام زهرا و شماره دانش‌آموزی ۱۲۳۹۲۳۴۳ است و چنانچه این فرد از جدول حذف شود، تمام ویژگی‌های مربوط به او نیز از جدول حذف

می‌شوند و تغییر در مشخصات این فرد تأثیری در مشخصات افراد دیگر نخواهد داشت لذا هر سطر از جدول ۱-۲ به عنوان یک رکورد در نظر گرفته می‌شود.

مدل‌های رابطه‌ای چند ویژگی مهم دارند:

- ردیف‌های تکراری در آن‌ها وجود ندارد.
- ترتیب ردیف‌ها اهمیتی در ساختار مدل ندارد.
- ترتیب ستون‌ها اهمیتی در ساختار مدل ندارد.
- مقادیر هر فیلد غیرقابل تجزیه است.

مفهوم کلید

• کلید در مدل رابطه‌ای، صفت یا ستونی است که برای هر کدام از رکوردها (سطرها) مقدار منحصر به فردی دارد. به عنوان مثال در جدول ۱-۲، ستون شماره دانش‌آموزی می‌تواند به عنوان فیلد کلید در نظر گرفته شود زیرا یقین داریم که برای هر رکورد منحصر به فرد است و مقدار این فیلد برای دو رکورد متفاوت از جدول یکسان نخواهد بود.

محیط عملیاتی

سازمان، مؤسسه یا نهادی را در نظر بگیرید که قصد داریم برای آن یک بانک اطلاعاتی ایجاد کنیم. در این صورت به این سازمان، مؤسسه یا نهاد، محیط عملیاتی گفته می‌شود. مثال‌های زیر نمونه‌هایی از محیط عملیاتی هستند:

۴ - یک اداره یا سازمان دولتی

۵ - یک بانک

۶ - یک کتابخانه

۱ - یک مدرسه

۲ - یک دانشگاه

۳ - یک بیمارستان



تمرین:

چند محیط عملیاتی نام ببرید.

در هر محیط عملیاتی با توجه به نیاز، تعدادی از اجزای آن به عنوان موجودیت انتخاب می‌شوند، به عنوان مثال در محیط مدرسه موجودیت‌هایی با عنوان دانش‌آموز، معلم و درس داریم. در محیط عملیاتی بیمارستان موجودیت‌های بیمار، پزشک، اتاق، بخش و ... وجود دارد.

تمرین:

چه موجودیت‌هایی برای محیط‌های عملیاتی بانک و کتابخانه می‌توان انتخاب کرد؟

محیط عملیاتی مدرسه را در نظر بگیرید، هر کدام از موجودیت‌های محیط عملیاتی ویژگی‌ها و صفات متعددی دارند که فقط برخی از آن‌ها را می‌توان به عنوان فیلد در نظر گرفت، به عنوان مثال یک دانش‌آموز می‌تواند صفات و ویژگی‌های زیر را داشته باشد:

- نام
- نام خانوادگی
- نام پدر
- نام مادر
- رنگ چشم
- قد
- وزن
- تاریخ تولد
- و ...

اما همان‌طور که مشاهده می‌کنید تمام این صفات در محیط عملیاتی مدرسه کاربردی نیستند، به عنوان مثال، رنگ چشم، قد، وزن، نام مادر و ... در محیط عملیاتی مدرسه به کار نمی‌روند. علاوه بر این در یک باشگاه ورزشی، یک بیمارستان یا هر محیط عملیاتی دیگر، با توجه به نیازها، صفات و ویژگی‌های متفاوتی در نظر گرفته می‌شوند.

به‌طور کلی در محیط عملیاتی مدرسه می‌توان ویژگی‌های زیر را برای هر کدام از موجودیت‌ها در نظر گرفت.

ویژگی‌های موجودیت دانش‌آموز

۱ - نام (name)

۲ - نام خانوادگی (family)

۳ - رشته تحصیلی (major)

۴ - تاریخ تولد (date)

۵ - محل تولد (area)

۶ - نمره (grade)

ویژگی‌های موجودیت معلم

۱ - نام (tname)

۲ - نام خانوادگی (tfamily)

۳ - آدرس (taddress)

۴ - تلفن (ttel)

ویژگی‌های موجودیت درس

۱ - کد درس (cnumber)

۲ - نام درس (cname)

۳ - تعداد ساعات در طول هفته (hcourse)

هر کدام از این موجودیت‌ها، یک جدول در محیط عملیاتی مدرسه و هر کدام از ویژگی‌ها و صفات موجودیت‌ها، یک فیلد در جدول‌های بانک اطلاعاتی خواهند بود. همان‌طور که پیش از این اشاره شد در مدل‌های رابطه‌ای نباید ردیف‌های تکراری وجود داشته باشد، برای روشن شدن مطلب کلاس خود را در نظر بگیرید، معلم شما برای صدا کردن دانش‌آموزان از کدام صفت دانش‌آموزان استفاده می‌کند، ممکن است پاسخ دهید نام خانوادگی، استفاده از نام خانوادگی برای صدا کردن دانش‌آموزان به این دلیل است که احتمال تکراری بودن کمتری دارد، تا اینجا می‌توان گفت نام خانوادگی به عنوان فیلد کلید در نظر گرفته شده است، اما اگر حتی یک مورد تکراری وجود داشته باشد و دو دانش‌آموز دارای نام خانوادگی مشابه باشند، به دلیل احتمال بروز اشتباه دیگر نمی‌توان از نام خانوادگی به تنهایی استفاده کرد، در این صورت معمولاً نام به دنبال نام خانوادگی استفاده می‌شود.

یعنی دو فیلد نام و نام خانوادگی به عنوان کلید استفاده می‌شوند.

• اگر کلید انتخاب شده شامل یک فیلد باشد، آن را فیلد کلید ساده می‌گویند.
 • اگر کلید انتخاب شده شامل دو یا چند فیلد باشد، آن را فیلد کلید مرکب می‌گویند.
 اما اگر قدری دقیق‌تر شویم، می‌بینیم که باز هم احتمال تکراری بودن نام و نام خانوادگی و این‌که دو نفر دارای نام و نام خانوادگی یکسان باشند نیز وجود دارد. اگر بخواهیم محیط عملیاتی مدرسه را طوری طراحی کنیم که احتمال تکراری بودن فیلدها را به صفر برسانیم بهتر است به جای ترکیب فیلدها یک فیلد جدید با نام شماره دانش‌آموزی به ویژگی‌های موجودیت دانش‌آموز اضافه کنیم، در این صورت اطمینان حاصل خواهد شد که هیچ دو دانش‌آموزی اطلاعات کاملاً یکسان ندارند و در نتیجه ایجاد سطر تکراری در جدول نخواهیم داشت.

برای تهیه جدول‌ها نکات زیر را در نظر بگیرید:

۱ - هر موجودیت مستقل را به عنوان یک جدول در نظر بگیرید.

• معلم

• درس

• دانش‌آموز

۲ - صفت‌های موجودیت‌ها به عنوان فیلدهای جدول‌ها در نظر گرفته می‌شوند.

جدول مربوط به دانش‌آموز

جدول ۲-۲ فیلدهای جدول دانش‌آموز (Student)

شماره دانش‌آموزی*	نام	نام خانوادگی	تاریخ تولد	محل تولد	نمره
(id)	(name)	(family)	(date)	(area)	(grade)

جدول مربوط به معلم

جدول ۲-۳ فیلدهای جدول معلم (teacher)

کد معلم*	نام	نام خانوادگی	آدرس	تلفن
tcode	tname	tfamily	taddress	ttel

جدول مربوط به درس

جدول ۴-۲ فیلدهای جدول درس (course)

تعداد ساعات در طول هفته	نام درس	کد درس*
hcourse	cname	cnumber

نکته



فیلدهایی که کنار آن‌ها علامت * درج شده است، فیلدهای کلیدی هستند.

در هر محیط عملیاتی موجودیت‌ها با یکدیگر مرتبط هستند. در طراحی و تحلیل بانک‌های اطلاعاتی، وجود این ارتباط و بررسی ماهیت آن، یکی از مراحل اصلی کار است. در محیط عملیاتی مدرسه، هر دانش‌آموز در سال تحصیلی در چند درس ثبت نام می‌کند. به کلمات «هر» و «چند» در ابتدای موجودیت‌های دانش‌آموز و درس دقت کنید، وجود این کلمات به این معناست که یک دانش‌آموز چند درس را انتخاب کرده است. ماهیت ارتباط دانش‌آموز-درس، «یک به چند» است.

۲-۲ انواع ماهیت ارتباط

با توجه به توضیحات بخش محیط عملیاتی، به تناظر بین تعداد موجودیت‌ها، ماهیت ارتباط گفته می‌شود و این تناظر می‌تواند انواع زیر را داشته باشد:

- ارتباط یک به یک (۱:۱)
- ارتباط یک به چند (۱:N)
- ارتباط چند به چند (N:N)

اگر در محیط عملیاتی مدرسه موجودیت‌های مدرسه و مدیر را نیز در نظر داشته باشیم، در ارتباط یک به یک، یک نمونه از موجودیت اول با یک نمونه از موجودیت دوم ارتباط دارد، به عنوان مثال هر مدرسه یک مدیر دارد.

در ارتباط یک به چند، یک نمونه از موجودیت اول با چند نمونه از موجودیت دوم ارتباط دارد و در مقابل هر نمونه از موجودیت دوم با یک نمونه از موجودیت اول ارتباط دارد، به عنوان مثال یک مدرسه چند معلم دارد و هر معلم در یک مدرسه تدریس می‌کند.

در ارتباط چند به چند، یک نمونه از موجودیت اول با چند نمونه از موجودیت دوم ارتباط دارد و همچنین یک نمونه از موجودیت دوم با چند نمونه از موجودیت اول مرتبط است.

مثال: در محیط عملیاتی مدرسه با در نظر گرفتن موجودیت‌های درس، معلم و دانش‌آموز، ماهیت‌های ارتباط و تناظر بین موجودیت‌ها به صورت زیر خواهد بود:

۱ - درس - معلم: هر معلم چند درس ارائه می‌دهد و هر درس توسط یک معلم ارائه می‌شود لذا نوع ارتباط بین درس و معلم یک به چند خواهد بود.

۲ - درس - دانش‌آموز: هر دانش‌آموز در هر دوره تحصیلی چند درس را انتخاب می‌کند و یک درس توسط چند دانش‌آموز اخذ می‌شود. لذا نوع ارتباط بین درس - دانش‌آموز چند به چند است.



تمرین:

نوع ارتباط معلم - دانش‌آموز را از لحاظ ماهیت ارتباط بررسی کنید.

۱-۲-۲ تأثیر ماهیت ارتباط بر بانک اطلاعات

ماهیت ارتباط بین موجودیت‌ها می‌تواند در طراحی جدول‌ها تأثیرگذار باشد و حتی ممکن است لازم شود، در انتخاب و طراحی جدول‌ها یک بازبینی کلی انجام شود.

برای ایجاد ساختار نهایی جدول‌های بانک اطلاعاتی باید نکات زیر را در نظر گرفت:

۱ - در ارتباط‌های ۱:۱ جدول‌ها در هم ادغام می‌شوند.

۲ - در ارتباط‌های ۱:N:۱ فیلد اصلی طرف ۱ به عنوان کلید خارجی به جدول طرف N افزوده می‌شود.

۳ - در ارتباط‌های N:N:۱ یک جدول جدید ساخته می‌شود، طوری که شامل کلیدهای هر دو طرف باشد.

۳-۲ محیط عملیاتی تولیدکنندگان قطعات صنعتی

به منظور ارائه مثال بیشتر، یک محیط عملیاتی جدید را در نظر می‌گیریم. مثلاً محیط عملیاتی تولیدکنندگان قطعات صنعتی در شهرهای مختلف. این محیط عملیاتی دارای دو

موجودیت اصلی «قطعه» و «تولیدکننده» است.

برای هر کدام از موجودیت‌ها می‌توان صفات (فیلدهای) زیر را در نظر گرفت:
موجودیت قطعه (جدول S)

- ۱ - شماره قطعه (S#) *
- ۲ - نام قطعه (Sname)
- ۳ - تعداد (Status)
- ۴ - شهر (City)

موجودیت تولیدکننده (جدول P)

- ۱ - شماره تولیدکننده (P #) *
- ۲ - نام تولیدکننده (Pname)
- ۳ - رنگ (Color)
- ۴ - وزن (Weight)
- ۵ - شهر (City)

هر قطعه ممکن است توسط چند تولیدکننده تولید شود و همچنین هر تولیدکننده، چند قطعه تولید می‌کند، در نتیجه ماهیت ارتباط این دو موجودیت چند به چند است. همان‌طور که پیش از این اشاره شد، در ارتباط‌های چند به چند، باید یک جدول جدید ساخته شود که فیلدهای آن حداقل باید شامل فیلدهای کلید اصلی دو جدول اولیه باشند. در نتیجه یک جدول جدید شامل دو موجودیت شماره قطعه و شماره تولیدکننده خواهیم داشت:

جدول (SP)

۱ - شماره قطعه (S#)

۲ - شماره تولیدکننده (P#)

در این محیط عملیاتی تعداد قطعه تولید شده نیز می‌تواند به جدول SP افزوده شود. (Qty) با توجه به توضیحات فوق سه جدول مربوط به این محیط عملیاتی به این صورت ترسیم می‌شوند:

جدول تولیدکننده (P)

جدول ۲-۵

شماره تولیدکننده (P#)	نام تولیدکننده (Sname)	تعداد خط تولید (Status)	شهر (City)
-----------------------	------------------------	-------------------------	------------

جدول قطعه (S)

جدول ۲-۶

شماره قطعه (S#)	نام قطعه (Sname)	رنگ (Color)	وزن (Weight)	شهر (City)
-----------------	------------------	-------------	--------------	------------

جدول تولیدکننده - قطعه (SP)

جدول ۲-۷

شماره قطعه (S#)	شماره تولیدکننده (P #)	تعداد قطعه تولید شده (Qty)
-----------------	------------------------	----------------------------

جداول فوق را با داده‌های نمونه زیر در نظر می‌گیریم:

جدول ۲-۸ تولیدکننده

P	P#	Pname	Status	City
	P1	احمدی	۲۰	تهران
	P2	رضایی	۱۰	کرمانشاه
	P3	سلیمی	۳۰	کرمانشاه
	P4	مهدوی	۲۰	تهران
	P5	میثمی	۳۰	شیراز

جدول ۲-۹ قطعه

S	S#	Sname	Color	Weight	City
	S1	پیچ	قرمز	۱۲	تهران
	S2	شیر	سبز	۱۷	کرمانشاه
	S3	لوله	آبی	۱۷	بوشهر
	S4	لوله	قرمز	۱۴	تهران
	S5	پیچ گوشتی	آبی	۱۲	کرمانشاه
	S6	فازمتر	قرمز	۱۹	تهران

جدول ۲-۱۰ تولیدکننده - قطعه

SP	S#	P#	Qty
	S1	P1	۳۰۰
	S1	P2	۲۰۰
	S1	P3	۴۰۰
	S1	P4	۲۰۰
	S1	P5	۱۰۰
	S1	P6	۱۰۰
	S2	P1	۳۰۰
	S2	P2	۴۰۰
	S3	P2	۲۰۰
	S4	P2	۲۰۰
	S4	P4	۳۰۰
	S4	P5	۴۰۰

برای این که با مفهوم جدول SP بیشتر آشنا شوید، سطر اول را در نظر بگیرید، مفهوم سطر اول این است که:
 قطعه S1 به میزان ۳۰۰ عدد توسط تولیدکننده P1، تولید شده است، یا این که تولیدکننده P1 از قطعه S1 به میزان ۳۰۰ عدد تولید کرده است.

تمرین: 

مفهوم سطرهای دوم و آخر را بیان کنید.

۲-۴ نرمال سازی بانک های اطلاعاتی

در محیط عملیاتی تولیدکنندگان قطعات صنعتی سه جدول S، P و SP وجود دارد، کاملاً مشخص است که در جدول های فوق فیلهای مربوط به هر کدام، کاملاً ضروری هستند و به درستی انتخاب شده اند.

جدول ۲-۱۱

S#	City	P#	Qty
S1	تهران	P1	۳۰۰
S1	تهران	P2	۲۰۰
S1	تهران	P3	۴۰۰
S1	تهران	P4	۲۰۰
S1	تهران	P5	۱۰۰
S1	تهران	P6	۱۰۰
S2	کرمانشاه	P1	۳۰۰
S2	کرمانشاه	P2	۴۰۰
S3	کرمانشاه	P2	۲۰۰
S4	تهران	P2	۲۰۰
S4	تهران	P4	۳۰۰
S4	تهران	P5	۴۰۰

فرض کنید، به جای جدول SP در بانک اطلاعاتی، جدول SCP را با چهار فیلد S#، City، P# و Qty طراحی کرده باشیم، در این صورت مقادیر نمونه ای جدول به این صورت خواهد بود:

با دقت در جدول ۲-۱۱ مشاهده می شود که داده های موجود در فیلد City کاملاً تکراری هستند و از طریق جدول های P و S نیز قابل دسترسی هستند، به این عمل تکراری بودن داده ها که موجب اشغال فضای بیشتر در پایگاه داده ها و جدول ها می شود، عمل افزونگی داده ها گفته می شود.

برای طراحی یک پایگاه داده استاندارد

باید دقت کرد که عمل افزونگی داده‌ها اتفاق نیفتد.

در طراحی یک بانک اطلاعاتی پس از تحلیل و بررسی موجودیت‌ها و در نظر گرفتن صفات هر کدام از آن‌ها لازم است جداول به شکل نرمال دربیابند، نرمال‌سازی (Normalization)، فرایندی است در رابطه با بانک‌های اطلاعاتی که با دو هدف عمده زیر انجام می‌شود:

کاهش افزونگی اطلاعات: به این معنی که اطلاعات فقط در یک مکان (جدول)

ذخیره شوند نه این‌که چند بار در جدول‌های مختلف تکرار شوند و در تمام بانک با استفاده از روابط (Relationship) تعریف شده قابل دسترسی باشند.

حفظ یکپارچگی اطلاعات: به این معنی که اعمال تغییرات روی اطلاعات

(نظیر ایجاد، بهنگام‌سازی و حذف) در یک مکان انجام شود و به دنبال آن آثار تغییرات در تمام بانک مشاهده گردد. برای روشن شدن مفهوم یکپارچگی به مثال زیر توجه نمایید:

فرض کنید یک بانک اطلاعاتی دارای دو موجودیت کتاب و نویسنده باشد. هر

یک از موجودیت‌های فوق دارای صفت‌های (Attribute) مختص به خود می‌باشند. به

عنوان نمونه موجودیت «کتاب» دارای صفت «نام نویسنده» و موجودیت «نویسنده» دارای

صفت‌های متعددی نظیر «نام نویسنده»، «آدرس نویسنده» و ... باشد. در صورتی که در

موجودیت «کتاب» یک رکورد ایجاد نماییم بدون این‌که نام نویسنده آن را در موجودیت

«نویسنده» ایجاد کرده باشیم، مسلماً یک ناهنجاری به وجود خواهد آمد.

با توجه به اهداف فوق می‌توان گفت که فرایند نرمال‌سازی از ناهنجاری‌های به

وجود آمده به دلیل بروز تغییرات در بانک اطلاعاتی جلوگیری خواهد نمود. با اعمال

فرایند نرمال‌سازی، یک بانک اطلاعاتی کارا و مطمئن خواهیم داشت. فرایند نرمال‌سازی،

فرم‌های متفاوتی دارد که انواع متداول آن به شرح ذیل می‌باشند:

• فرم اول نرمال (NF۱)

• فرم دوم نرمال (NF۲)

• فرم سوم نرمال (NF۳)

۱-۴-۲ فرم اول نرمال (NF۱)

موجودیت یا جدولی در فرم اول نرمال است که هیچ‌کدام از فیلدهای آن قابل تقسیم نباشد،

به عنوان مثال در محیط عملیاتی مدرسه موجودیت دانش‌آموز را در نظر بگیرید، چنانچه جدول مربوط به آن را به صورت زیر طراحی کرده باشید:

جدول ۱۲-۲

نام و نام خانوادگی	نام پدر	تاریخ تولد	محل تولد	معدل
--------------------	---------	------------	----------	------

حال جدول فوق با مقادیر نمونه‌ای به صورت زیر خواهد بود:

جدول ۱۳-۲

نام و نام خانوادگی	نام پدر	تاریخ تولد	محل تولد	معدل
مینا رضایی	احمد	۷۵/۳/۱	تهران	۱۹/۳۰
مریم توکلی	محمد رضا	۷۴/۷/۲	تهران	۱۹/۰۲
سارا بهرامی	مهدی	۷۵/۶/۳	مشهد	۱۸/۲۵
زهرا ساداتی	محمد سعید	۷۴/۱۲/۵	کرمانشاه	۱۸/۱۷
فاطمه خدابخشی	علی اکبر	۷۵/۶/۶	تهران	۱۹/۳۵

مقادیر فیلد اول جدول قابل تجزیه هستند به این معنا که اگر هر کدام از اسامی نام و نام خانوادگی را تقسیم کنیم، آنگاه دو بخش اطلاعاتی معنادار خواهیم داشت، به عنوان نمونه در رکورد اول، فیلد اول دارای مقدار «مینا رضایی» است، می‌توان در این فیلد دو مقدار «مینا» و «رضایی» را استخراج نمود طوری که هر کدام از این اطلاعات معنادار باشند (جزء اول نام و جزء دوم، نام خانوادگی رکورد اول است). در این حالت می‌توان گفت که جدول در فرم نرمال اول نیست، زیرا مقادیر فیلدها قابل تجزیه هستند.

سؤال: فیلد تاریخ تولد نیز به اجزای روز، ماه و سال قابل تجزیه است، آیا در این حالت نیز می‌توان گفت جدول در حالت نرمال نوع اول قرار ندارد؟

پاسخ: اصطلاح معنادار وابسته به نوع محیط عملیاتی است، در محیط عملیاتی مدرسه، فیلد تاریخ به عنوان یک فیلد مجزا و غیر قابل تجزیه در نظر گرفته می‌شود. همین مسأله را می‌توان در فیلد شماره تلفن نیز بررسی کرد، هر شماره تلفن از ترکیب چند عدد در کنار هم تشکیل شده است، اگر اعداد را تجزیه کنیم، معنایی در محیط عملیاتی نخواهند داشت. در نتیجه در جدول ۱۳-۲ فقط فیلد اول قابل تجزیه است.

۲-۴-۲ فرم دوم نرمال (NF ۲)

برای روشن شدن مفهوم فرم نرمال باید مفهوم وابستگی تابعی را بشناسید، مفهوم وابستگی تابعی در بانک‌های اطلاعاتی مشابه مفهوم تابع در ریاضی است.

مفهوم تابع در ریاضی: در یک رابطه با نام R ، متغیر X با Y وابستگی تابعی دارد، اگر به ازای هر مقدار X فقط و فقط یک مقدار Y وجود داشته باشد، در این صورت وابستگی تابعی X و Y به صورت $X \rightarrow Y$ شان داده می‌شود.

جدول ۲-۱۴

X	Y	Z
x1	y1	z1
x2	y2	z3
x1	y1	z2
x3	y2	z2
x4	y3	z3

رابطه R را مطابق جدول ۲-۱۴ برای سه متغیر X

و Y و Z در نظر بگیرید:

در رابطه R ، می‌خواهیم وابستگی تابعی $X \rightarrow Y$ و $Y \rightarrow Z$ را بررسی کنیم: وابستگی تابعی $X \rightarrow Y$ برقرار است، زیرا به ازای مقادیر یکسان از $x1$ ، مقادیر یکسانی در $y1$ وجود دارد، به عنوان مثال ستون X را در نظر بگیرید، این ستون دارای دو مقدار تکراری $x1$ است، که هر دو مقدار آن با $y1$ از ستون Y متناظر است، در نتیجه به دلیل مساوی بودن طرف دوم رابطه، وابستگی تابعی $X \rightarrow Y$ برقرار است. وابستگی تابعی $Y \rightarrow Z$ برقرار نیست، ستون Y دارای دو مقدار تکراری $y1$ و $y2$ است، مقدار $y1$ متناظر با $z1$ و $z2$ در ستون Z است، لذا به دلیل عدم تساوی این دو مقدار می‌توان گفت وابستگی تابعی $Y \rightarrow Z$ برقرار نیست.

نکته برای بررسی برقراری وابستگی تابعی، حتی اگر یک نمونه یافت شود که وابستگی تابعی را نقض می‌کند، باید گفت که وابستگی تابعی برقرار نیست.

با توجه به مفهوم وابستگی تابعی می‌توان مفهوم فرم دوم نرمال را بررسی کرد: موجودیتی در فرم دوم نرمال است که اولاً در فرم نرمال اول باشد، ثانیاً تمام فیلدهای آن وابستگی تابعی را با کلید اصلی نقض نکنند.



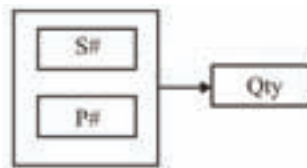
نکته اگر جدول دارای یک فیلد کلید ساده باشد، به دلیل ویژگی‌های فیلد کلید می‌توان گفت که جدول در فرم دوم نرمال قرار دارد، اما اگر یک جدول دارای فیلد کلید اصلی مرکب باشد، به این سادگی نمی‌توان نتیجه گرفت که جدول در حالت فرم دوم نرمال قرار دارد یا خیر.

جدول‌های مربوط به محیط عملیاتی تولیدکنندگان قطعات را در نظر بگیرید (جدول‌های ۸-۲، ۹-۲ و ۱۰-۲).

در جدول‌های S و P به دلیل نرمال نوع اول بودن و وجود کلید اصلی ساده می‌توان به راحتی نتیجه گرفت که رابطه مربوط به آن‌ها در حالت نرمال دوم قرار دارد.

اما در جدول SP، کلید اصلی شامل ترکیبی از فیلدهای S# و P# است، در نتیجه برای بررسی این‌که آیا در حالت نرمال دوم قرار دارد یا خیر، باید آن را از نظر وابستگی تابعی بررسی کرد.

نمودار وابستگی تابعی مربوط به جدول SP را می‌توان به صورت زیر نشان داد:



شکل ۴-۲

همان‌طور که می‌دانید جدول SP از روی جداول S و P استخراج شده است، با دقت در جدول SP (جدول ۱۰-۲) می‌بینیم که اجزای کلید اصلی (S# و P#) به‌طور مجزا با Qty وابستگی تابعی ندارند و وابستگی تابعی فقط توسط ترکیب S# و P# وجود دارد، لذا می‌توان گفت که رابطه جدول SP نیز در حالت نرمال دوم NF قرار دارد.

مثال: فرض کنید در محیط عملیاتی تولیدکنندگان قطعات جدول ۱۵-۲ را به جای

SP در نظر گرفته‌ایم:

جدول ۱۵-۲

S#	Status	City	P#	Qty
S1	۲۰	تهران	P1	۳۰۰
S1	۲۰	تهران	P2	۲۰۰
S1	۲۰	تهران	P3	۴۰۰
S1	۲۰	تهران	P4	۲۰۰
S1	۲۰	تهران	P5	۱۰۰
S1	۲۰	تهران	P6	۱۰۰
S2	۱۰	کرمانشاه	P1	۳۰۰
S2	۱۰	کرمانشاه	P2	۴۰۰
S3	۱۰	کرمانشاه	P2	۲۰۰
S4	۲۰	تهران	P2	۲۰۰
S4	۲۰	تهران	P4	۳۰۰
S4	۲۰	تهران	P5	۴۰۰

این جدول در فرم نرمال اول قرار دارد اما در فرم نرمال نوع دوم قرار ندارد. اگر فیلدهای P# و S# را به عنوان کلید اصلی مرکب در نظر بگیریم، بخش‌هایی از کلید اصلی با سایر فیلدها وابستگی تابعی دارد. S# به تنهایی با City وابستگی تابعی دارد که این امر با تعریف فرم دوم نرمال در تناقض است و نباید اجزای کلید اصلی در یک رابطه با سایر فیلدها وابستگی تابعی داشته باشد.

۳-۴-۲ فرم سوم نرمال (۳ NF)

موجودیت یا جدولی در فرم سوم نرمال است که اولاً در فرم دوم نرمال بوده، ثانیاً تمام صفت‌های غیرکلید آن وابستگی تابعی فقط به کلید اصلی داشته باشند نه به یک صفت غیرکلید، یا به عبارت دیگر صفات غیرکلید وابستگی تابعی با واسطه با کلید اصلی نداشته باشند به عنوان نمونه فرض کنید در محیط عملیاتی تولیدکنندگان قطعات، این جدول‌ها را در نظر گرفته‌ایم:

جدول ۲-۱۶

S#	City
S1	تهران
S2	کرمانشاه
S3	کرمانشاه
S4	تهران
S5	بوشهر

جدول ۲-۱۷

City	Status
بوشهر	۳۰
تهران	۲۰
کرمانشاه	۱۰
شیراز	۵۰

در این دو جدول می‌توان نمودار وابستگی تابعی را به صورت زیر ترسیم کرد:

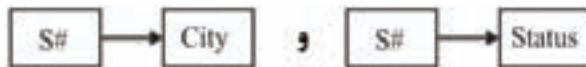


شکل ۲-۵

به دلیل وجود واسطه در وابستگی تابعی، جدول‌های ۲-۱۶ و ۲-۱۷ در فرم سوم نرمال نیستند.

(در این جدول‌ها کلید S# با City وابستگی تابعی دارد و City نیز با Status وابستگی تابعی دارد).

برای حذف این وابستگی تابعی می‌توان فیلد واسطه را در وابستگی تابعی حذف کرد:



شکل ۲-۶

در این صورت وابستگی تابعی با واسطه حذف شده است، بنابراین رابطه در حالت سوم نرمال قرار دارد.

واژه‌نامه

Attribute	صفت
Entity	موجودیت
Hierarchical	سلسله مراتبی
Network	شبکه
Normalization	نرمال‌سازی
Producer	تولیدکننده
Relational	رابطه‌ای

خلاصه مطالب

- پایگاه داده مجموعه‌ای سازمان‌یافته از اطلاعات است که می‌تواند به صورت رکوردهای ذخیره شده در جداول باشد و با یک روش اصولی و از طریق فرم‌ها، گزارش‌ها و ... می‌تواند به درخواست‌های کاربران پاسخ دهد.
- مدل‌های پایگاه داده طرح و الگوی کاری برای کاربران پایگاه داده‌ها را در سطح منطقی مشخص می‌کند.
- مدل سلسله مراتبی توسط گراف پیاده‌سازی می‌شود که ارتباط بین گره‌ها از بالا به پایین صورت می‌گیرد و مسیری از گره‌های پایین‌تر به گره‌های بالاتر وجود ندارد.
- مدل شبکه‌ای شباهت زیادی به سلسله مراتبی دارد، اما به دلیل وجود ارتباط از پایین به بالا پیاده‌سازی آن راحت‌تر است.
- مدل رابطه‌ای یکی از بهترین و سریع‌ترین مدل‌های پایگاه داده است و اطلاعات در قالب جدول‌های مجزا مشخص می‌شوند.
- مدل رابطه‌ای بهترین مدل پیاده‌سازی سیستم‌های بانک‌های اطلاعاتی است. در این مدل پایگاه داده از دید کاربران شامل یک مجموعه جدول است که هر جدول دارای سطرها و ستون‌هایی است.
- تعیین ماهیت ارتباط در مدل رابطه‌ای یکی از مهم‌ترین مراحل طراحی بانک اطلاعاتی است و پس از تحلیل موجودیت‌ها و بررسی ماهیت ارتباط آن‌ها، باید رابطه‌ها را به صورت

جدول پیاده‌سازی کرد.

- پس از تهیه جدول‌های بانک اطلاعاتی به منظور بهینه‌سازی جداول، کاهش افزونگی اطلاعات و حفظ یکپارچگی اطلاعات باید آن‌ها را نرمال‌سازی کرد.
- نرمال‌سازی مدل‌های مختلفی دارد که مهم‌ترین آن‌ها عبارتند از:
 - فرم نرمال اول
 - فرم نرمال دوم
 - فرم نرمال سوم

آزمون نظری

- ۱ - به مجموعه‌ای سازمان‌یافته از اطلاعات گفته می‌شود.
 - الف - پایگاه داده - ب - جدول - ج - رکورد - د - فیلد
- ۲ - پیاده‌سازی کدام یک از مدل‌های پایگاه داده زیر مشکل‌تر از سایر مدل‌هاست؟
 - الف - شبکه‌ای - ب - رابطه‌ای - ج - شیء‌گرا - د - سلسله‌مراتبی
- ۳ - مجموعه‌ها و رکوردها در کدام مدل بررسی می‌شود؟
 - الف - شبکه‌ای - ب - رابطه‌ای - ج - شیء‌گرا - د - سلسله‌مراتبی
- ۴ - در سیستم عملیاتی مدرسه موجودیت‌های دانش‌آموز - درس چه نوع رابطه‌ای با هم دارند؟
 - الف - یک به یک - ب - یک به چند
 - ج - چند به چند - د - این دو موجودیت به‌طور مستقیم رابطه‌ای ندارند.
- ۵ - کدام یک از گزینه‌های زیر از اهداف نرمال‌سازی بانک اطلاعاتی است؟
 - الف - کاهش افزونگی و حفظ یکپارچگی اطلاعات
 - ب - افزایش افزونگی و افزایش یکپارچگی اطلاعات
 - ج - کاهش تعداد جدول‌های بانک اطلاعاتی و حفظ یکپارچگی
 - د - افزایش تعداد رابطه‌های بانک اطلاعاتی و افزایش یکپارچگی
- ۶ - اگر تمام صفت‌های غیر کلیدی یک موجودیت به تمام کلید اصلی وابستگی تابعی داشته باشند و همچنین تمام صفت‌های آن یکتا باشد، این موجودیت در کدام فرم نرمال است؟
 - الف - فرم نرمال اول - ب - فرم نرمال دوم
 - ج - فرم نرمال سوم - د - هیچ‌کدام

آشنایی با دستورات کار با پایگاه داده

واحد کارسوم

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

- ۱ - با اصول ایجاد پایگاه داده در MySQL آشنا باشد.
- ۲ - بتواند یک جدول جدید در MySQL ایجاد کند.
- ۳ - بتواند یک جدول را در MySQL حذف کند.
- ۴ - ساختار جداول را اصلاح کند.
- ۵ - بتواند جدول‌ها را تغییر نام دهد.
- ۶ - بتواند یک رکورد و داده‌های جدید به جدول اضافه کند.
- ۷ - بتواند اطلاعات را از جداول حذف کند.
- ۸ - بتواند اطلاعات جداول‌های موجود را ویرایش و به‌روز رسانی کند.
- ۹ - بتواند اطلاعات جدول‌ها را انتخاب کند.

زمان(ساعت)	
عملی	نظری
۳۵	۱۷

مقدمه

همان طور که می‌دانید سیستم‌های مدیریت پایگاه داده مختلفی وجود دارند که با توجه به نوع نیاز، اندازه سیستم، بودجه و ... می‌توان یکی از آن‌ها را برای پیاده‌سازی سیستم بانک اطلاعاتی محیط عملیاتی استفاده کرد، سیستم‌های مدیریت پایگاه داده معمولاً از یک زبان ساخت‌یافته استاندارد با نام SQL به‌طور مشترک استفاده می‌کنند.

۳-۱ آشنایی با SQL

- SQL مخفف عبارت Structured Query Language (زبان جستجوی ساخت‌یافته) است و به کاربر امکان اتصال و دسترسی به اطلاعات موجود در یک پایگاه داده را می‌دهد.
- زبان SQL قادر است تا برای یک پایگاه داده عمل جستجو و گزینش اطلاعات را انجام دهد و همچنین اطلاعات ذخیره شده در یک پایگاه داده را بازیابی، حذف، ذخیره، اضافه یا به روزرسانی کند.
- زبان SQL یک استاندارد بین‌المللی است.

Table ۳-۱-۱

با مفهوم جدول در واحدکار دوم آشنا شده‌اید. جدول‌ها مهم‌ترین عناصر پایگاه داده هستند که برای ذخیره و نگهداری سازمان‌یافته اطلاعات مورد استفاده قرار می‌گیرند.

Query ۳-۱-۲

از Query برای ایجاد یک پرسش و جستجو در جدول و استخراج اطلاعات مورد نظر استفاده می‌شود. نتایج حاصل از یک Query در یک جدول موقت که دارای فیلدها و سطرهای واجد شرایط جستجو است، نمایش داده می‌شود.

مثال: به عنوان مثال می‌خواهیم، مشخصات دانش‌آموزانی را که در ترم دوم سال تحصیلی ۸۵-۸۶ موفق به گذراندن درس پایگاه داده شده‌اند از جدول دانش‌آموزان استخراج کنیم. برای این منظور از یک Query استفاده می‌کنیم. عمده دستورات SQL برای ایجاد و مدیریت

Query استفاده می‌شوند. در مباحث بعدی، کار با یک Query را بررسی می‌کنیم.

۳-۱-۳ SQL Data Manipulation Language (DML) یا زبان دستکاری

اطلاعات SQL

این بخش از دستورات SQL، شامل دستوراتی برای به روز کردن، ورود اطلاعات یا حذف رکوردها در محیط پایگاه داده است. عمده دستورات این بخش عبارتند از:

- Select**: برای استخراج اطلاعات از یک جدول بانک اطلاعاتی مورد استفاده قرار می‌گیرد.
- Update**: برای به روز کردن اطلاعات یک جدول بانک اطلاعاتی مورد استفاده قرار می‌گیرد.
- Delete**: برای حذف اطلاعات از یک جدول بانک اطلاعاتی مورد استفاده قرار می‌گیرد.
- Insert**: برای ورود اطلاعات به یک جدول بانک اطلاعاتی مورد استفاده قرار می‌گیرد.

۳-۱-۴ SQL Data Definition Language (DDL) یا زبان تعریف اطلاعات SQL

از این قسمت می‌توان برای ایجاد یا حذف جداول از بانک اطلاعاتی، تعریف اندیس‌ها، کلیدواژه‌ها، ایجاد ارتباط و پیوند بین جدول‌ها یا ایجاد محدودیت‌های دسترسی استفاده کرد. عمده دستورات این بخش عبارتند از:

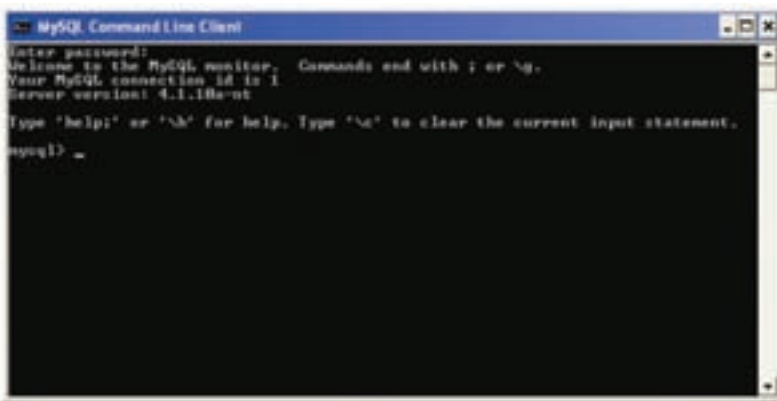
- Create Table**: برای ایجاد یک جدول جدید به کار می‌رود.
- Alter Table**: برای تغییر اطلاعات موجود در یک جدول به کار می‌رود.
- Drop Table**: برای حذف یک یا چند جدول از بانک اطلاعاتی به کار می‌رود.
- Create Index**: برای ایجاد اندیس یا کلید واژه در یک جدول به کار می‌رود.
- Drop Index**: برای حذف اندیس یا کلید واژه در یک جدول به کار می‌رود.

اندیس (Index): اندیس عبارت است از یک شماره که به هر یک از فیلدها در سطرهای یک جدول اختصاص داده می‌شود. اندیس‌ها از دید کاربر کاملاً مخفی هستند. استفاده از اندیس‌ها باعث می‌شود تا برنامه بتواند مقادیر سطرهای مختلف را بر حسب مقدار یک فیلد و بر حسب شماره اندیس آن‌ها از کم به زیاد یا برعکس مرتب کند و در عملیات جستجو باعث بالا رفتن سرعت جستجو می‌شود.

برای اجرای دستورات SQL و مشاهده نتیجه آن باید از یک سیستم مدیریت پایگاه داده استفاده کرد. سیستم مدیریت پایگاه داده‌ای که در این کتاب استفاده شده است، MySQL می‌باشد.

نکته در انتهای هر دستور SQL باید از علامت ; استفاده شود.

ابتدا از نصب MySQL روی سیستم خود مطمئن شوید، با نصب MySQL در واحدکار اول آشنا شده‌اید. سپس تمام دستورات را از طریق command line اجرا کنید. برای ورود به محیط command line از منوی Start گزینه All Programs سپس MySQL، MySQL Server 5.1 و در نهایت MySQL Command Line Client را انتخاب نمایید.



شکل ۳-۱

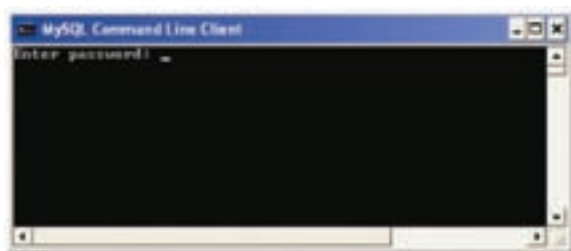
اگر رمز عبوری را هنگام نصب تعیین کرده‌اید، آن را وارد کنید، سپس کلید Enter را فشار دهید و اگر هیچ رمز ورودی تعیین نشده است، بدون وارد کردن هیچ کاراکتری، کلید Enter را فشار دهید.

۳-۲ دسترسی به MySQL از طریق خط فرمان

اگر قصد داشته باشید از طریق خط فرمان با محیط MySQL کار کنید، ابتدا باید MySQL را روی سیستم نصب کنید (با شیوه نصب MySQL در واحد کار اول آشنا شدید). از منوی Start و در قسمت All Programs گزینه MySQL، سپس MySQL Server و بعد از آن MySQL Command Line Client را انتخاب کنید.

در پنجره شکل ۳-۲ اگر کلمه رمزی را هنگام نصب MySQL تعیین کرده‌اید آن را وارد کنید، سپس کلید Enter را فشار دهید، در غیر این صورت بدون وارد کردن هیچ

کاراکتری کلید Enter را فشار دهید.



شکل ۳-۲

مقابل عبارت `MySQL>` علامت چشمک‌زن ظاهر می‌شود که به این معناست که به `MySQL` وارد شده‌اید و می‌توانید دستورات لازم را اجرا کنید. پس از ورود به محیط `MySQL` اگر بخواهید اعمال مختلف را روی یک بانک اطلاعاتی انجام دهید، ابتدا باید با استفاده از دستور `use` پایگاه داده مورد نظر را انتخاب کنید. ساختار این دستور به شکل زیر است:

نام پایگاه داده `use`

مثال: `use db1;`

۳-۳ دستور `create`

دستور `Create` برای ایجاد یک جزء جدید در سیستم‌های پایگاه داده مورد استفاده قرار می‌گیرد.

۳-۳-۱ ایجاد یک پایگاه داده (`Create a DataBase`)

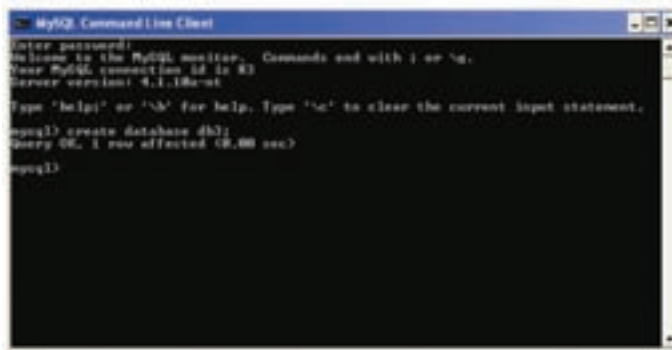
برای ایجاد یک پایگاه داده جدید از این دستور استفاده می‌شود:

نام پایگاه داده `create DataBase`

مثال: `create DataBase UniverArea ;`

مثال: `create DataBase db3 ;`

برای اجرای دستور فوق در خط فرمان، دستور مورد نظر را وارد کنید، سپس کلید `Enter` را فشار دهید.



شکل ۳-۳

پیغام Query OK به این معناست که درخواست با موفقیت انجام پذیرفته است.

۳-۳-۲ ایجاد یک جدول (create Table)

برای ایجاد یک جدول ۳ عامل اصلی را باید در نظر گرفت:

تعیین یک نام منحصر به فرد برای جدول

تعیین تعداد ستون‌های (فیلدها) جدول و نام آن‌ها

تعیین نوع داده‌ای ستون‌های جدول و اندازه آن‌ها در صورت نیاز

شکل کلی دستور ایجاد جدول در پایگاه داده به صورت زیر است:

```
create Table Students  
(  
StudentID Numeric (8) ,  
Name Char (15) ,  
Family Char (15) ,  
... ;  
)
```

نکته نوع داده انتخابی برای ستون‌ها، باید یکی از انواع داده‌های مجاز در SQL باشد. برای فیلدهایی که می‌خواهید حداکثر طول مجاز را تعیین کنید، می‌توانید مقدار آن را در پرانتز جلوی نام نوع داده فیلد ذکر کنید.



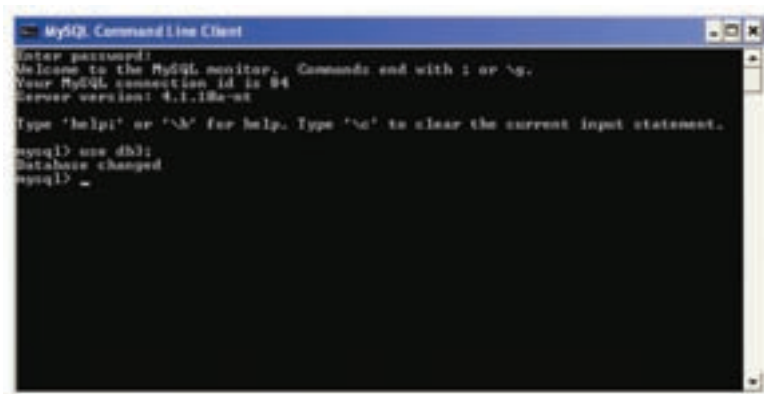
مثال: نوع داده‌ای از نوع کاراکتر با حداکثر طول ۱۵:

Name Char (15) ;

برای ایجاد جدول توسط خط فرمان، در صورتی که پایگاه داده را قبلاً ایجاد کرده باشید، ابتدا باید با استفاده از دستور

use نام پایگاه داده ;

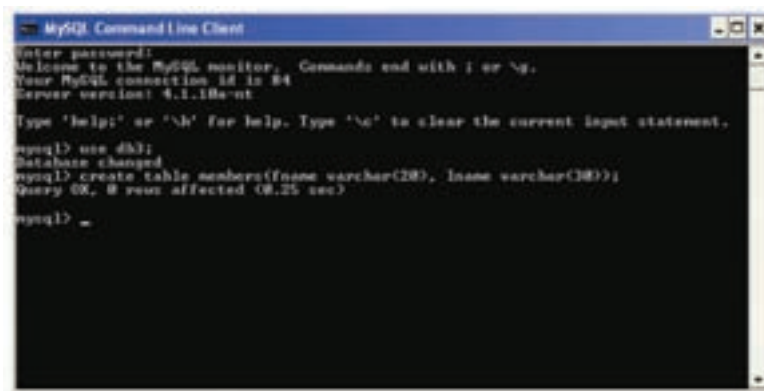
مشخص نمایید که می‌خواهید روی کدام پایگاه داده کار کنید.



شکل ۳-۴

حال برای ایجاد جدول باید دستور ایجاد جدول به همراه فیلدها و نوع هر کدام را به صورت زیر درج کنید:

(... و نوع متغیر دوم فیلد دوم و نوع متغیر اول فیلد اول) نام جدول create table



شکل ۳-۵

سایر دستورات SQL را نیز میتوانید از طریق خط فرمان اجرا کنید.

۳-۳-۳ ایجاد یک اندیس (create Index)

از این دستور برای ایجاد اندیس در ستون‌های جدول استفاده می‌شود. در یک جدول می‌توان برای یک یا چند ستون جدول، اندیس ایجاد کرد که این اندیس‌ها باعث بالا رفتن سرعت جستجو در رکوردهای جدول می‌شوند. اندیس یک شماره است که به هر سطر جدول اختصاص داده می‌شود و معمولاً از صفر شروع می‌شود. اندیس‌ها از دید کاربر مخفی هستند و هر اندیس یک نام منحصر به فرد دارد. شکل کلی ایجاد یک اندیس به صورت زیر است:

نام اندیس create Index

(نام فیلد مورد نظر) نام جدول on

مثال: create Index Rowup

on Students (StudentID) ;

نکته آوردن واژه Unique قبل از واژه Index در دستور قبل، باعث ایجاد یک شماره منحصر به فرد برای هر رکورد در ستون جدول می‌شود، به این معنی که هیچ دو رکوردی دارای اندیس‌های یکسان نخواهند بود.

- اندیس‌ها به صورت پیش‌فرض صعودی (از کم به زیاد) هستند، برای ایجاد یک اندیس با ترتیب برعکس (از زیاد به کم) باید بعد از نام فیلد در پرانتز، عبارت DESC ذکر شود.
- به منظور ایجاد یک اندیس برای بیش از یک فیلد، باید نام فیلدهای مورد نظر را به ترتیب در پرانتز بعد از نام جدول وارد کرد.

مثال: برای ایجاد اندیس روی فیلدهای StudentID و Name و Family، در جدول ۳-۱،

کد SQL را می‌توان به صورت زیر نوشت:

```
create Index Rowup
```

```
on Students (StudentID , Name , Family) ;
```

جدول ۳-۱

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	۱۸
10724113	Ehsan	Amiri	Electronic	Karaj	۱۴
10254861	Zahra	Hoseini	Computer	Tehran	۱۷
27365187	Sahar	Ahmadi	Electronic	Bam	۱۶
35654415	Hesam	Razavi	Electronic	Tehran	۱۹



جدول ۳-۱ را در MySQL ایجاد کنید.

۳-۴ دستور select

دستور Select برای انتخاب و استخراج اطلاعات مورد نظر از یک یا چند جدول و سپس مشاهده نتایج در یک جدول موقت استفاده می‌شود. شکل کلی این دستور به شکل زیر است:

```
select نام ستون یا ستون‌های مورد نظر
from نام جدول;
```

مثال: برای مشاهده ستون‌های Name و Family از جدول ۳-۱، دستور SQL به صورت

زیر نوشته می‌شود:

کد:

```
select Name,Family
from Student
```

خروجی

Name	Family
Ahamd	Rezaee
Ehsan	Amiri
Zahra	Hoseini
Sahar	Ahmadi

نکته برای انتخاب تمام ستون‌های جدول می‌توان از نشانگر * در قسمت نام ستون استفاده کرد. در این صورت تمام ستون‌های جدول در خروجی نمایش داده خواهند شد:

```
select *  
from Student;
```

تمرین

۱ - یک دستور SQL بنویسید که فیلدهای نام، نام خانوادگی و نمره جدول ۳-۱ را نشان دهد.

۲: جدول‌های ۲-۸، ۲-۹ و ۲-۱۰ (واحدکار دوم) را در محیط MySQL ایجاد کنید، سپس تمام فیلدهای این سه جدول را در خروجی نمایش دهید.

واژه کلیدی DISTINCT: چنانچه در ستون‌های مورد جستجو، موارد تکراری وجود داشته باشد در نتیجه خروجی نمایش داده خواهند شد. برای جلوگیری از چنین موردی و عدم نمایش موارد تکراری، پس از دستور Select عبارت DISTINCT نوشته می‌شود:

جدول ۳-۲

Orders Table	
Order Number	Company
1012	Dell
1014	Sega
1016	Dell
1018	Sony

مثال ۱:

کد:

```
select DISTINCT Company  
from Orders;
```

خروجی

Company
Dell
Sega
Sony

مثال ۲:

کد:

```
select Company
from Orders;
```

خروجی

Company
Dell
Sega
Dell
Sony

۵-۳ دستور where

دستور where برای افزودن شرط یا شرط‌هایی برای محدود کردن نتایج جستجو و استخراج نتایج دقیق‌تر استفاده می‌شود. این دستور باید پس از دستور select و تعیین ستون‌ها از جدول مورد نظر به کار رود. با استفاده از عملگرهای and، or و پرانتز می‌توان چندین شرط را با هم ترکیب کرد. برنامه فقط سطرهایی از جدول را که با شرط‌های ارائه شده سازگاری دارند نمایش خواهد داد. این دستور با عملگرهای زیر می‌تواند به کار رود:

جدول ۳-۳

عملگر	مفهوم	عملگر	مفهوم
=	برابر با، مساوی	!=	نا برابر با، نامساوی
<	کوچک‌تر یا کمتر	<=	کوچک‌تر مساوی
>	بزرگ‌تر یا بیشتر	>=	بزرگ‌تر مساوی
Between	بین دو مقدار	like	تعیین یک قالب برای جستجو

مثال: انتخاب نام و نام خانوادگی دانش‌آموزانی که شهر محل تولد آن‌ها تهران است از جدول ۳-۱ (Student) به این صورت انجام می‌شود:

کد:

```
select Name , Family
from Student
where Area = "Tehran ";
```

خروجی

Name	Family
Ahmad	Rezaee
Zahra	Hoseini
Hesam	Razavi

۵-۳ اعمال بیش از یک شرط و ادغام شرطها با عملگر and

مثال: مشخصات نام و نام خانوادگی دانش‌آموزانی که شهر آنها تهران بوده و نمره آنها بیش از پانزده است:
کد:

select Name , Family
from Student
where Area = " Tehran"
and Grade > 15;

خروجی

Name	Family
Ahmad	Rezaee
Hesam	Razavi

نکته در هنگام اعمال شرط جستجو، مقادیر رشته‌ای (متن) باید بین دو علامت " " قرار بگیرند، ولی در مورد موارد عددی این کار لازم نیست.

تمرین: 

در جدول S که آنرا در تمرین ۲ ایجاد کردید، اطلاعات رکوردهایی را نمایش دهید که در شهر تهران تولید شده‌اند.

۵-۳ عبارت like

این عبارت برای تعیین یک قالب خاص برای جستجو به کار می‌رود:
مثال: نام دانش‌آموزانی که شهر محل تولد آنها با حرف K شروع می‌شود:

select Name from Student
where area like "K%";

مثال: نام دانش‌آموزانی که نام شهر محل تولد آنها به an ختم می‌شود:
select Name from Student

where Area like "%an";

مثال: نام دانش‌آموزانی که شهر محل تولد آنها دارای حروف st است:

Select Name From Student

Where Area like "%st%";

مثال: نام دانش آموزانی که شهر محل تولد آن‌ها ۶ حرفی بوده و با حرف T شروع

می‌شود:

```
select Name from Student
```

```
where Area like "T-----" ;
```

نتیجه‌گیری: علامت % به معنای هر چند کاراکتر یا حرف دلخواه در قالب ارائه شده

و علامت - نمایانگر یک کاراکتر دلخواه است.

۳-۶ دستور insert

این دستور برای وارد کردن و ایجاد یک رکورد جدید در جدول استفاده می‌شود.

شکل کلی این دستور به این شرح است:

```
insert into نام جدول
```

```
values (...و مقدار ۲ و مقدار ۱) ;
```

می‌توان تعیین کرد که مقادیر مورد نظر به ترتیب به کدام ستون‌های جدول وارد شوند.

برای این منظور ابتدا نام ستون‌های مورد نظر را به ترتیب جلوی نام جدول در یک پرانتز

وارد کرده و با کاما از هم جدا می‌کنیم. سپس مقادیر متناظر را به همان ترتیب پس از واژه

values در پرانتز وارد می‌کنیم.

نکته چنانچه برای فیلد یا فیلدهایی مقداری در نظر گرفته نشود، جای آن‌ها در

جدول خالی می‌ماند. فقط باید به ترتیب نام ستون‌ها و مقادیر دقت کرد.

در شکل ۳-۶ درج داده در جدول members را مشاهده می‌کنید.

حال برای اینکه اطلاعات درج شده را مشاهده کنید، می‌توانید از دستور زیر استفاده

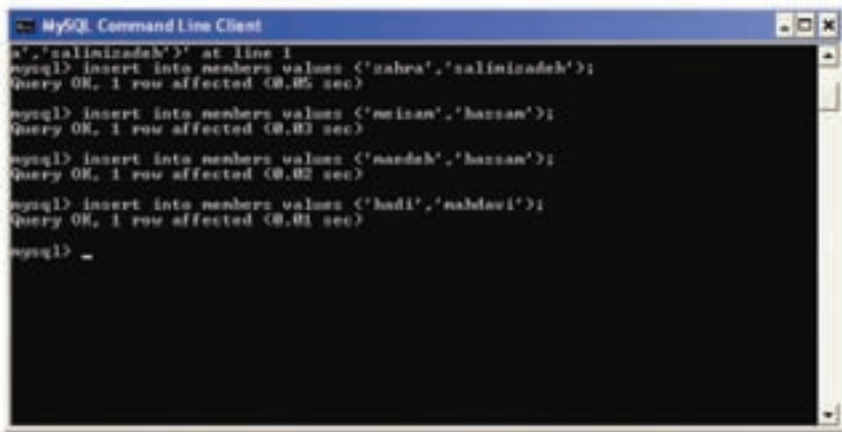
نمایید:

```
SELECT
```

```
* from members;
```

این دستور را نوشته، سپس کلید Enter را فشار دهید. مشاهده می‌کنید اطلاعاتی که قبلاً

در جدول وارد شده بود مطابق شکل ۳-۷ ظاهر شده‌اند.



```
mysql> insert into members values ('zahra','salimizadeh');
Query OK, 1 row affected (0.05 sec)

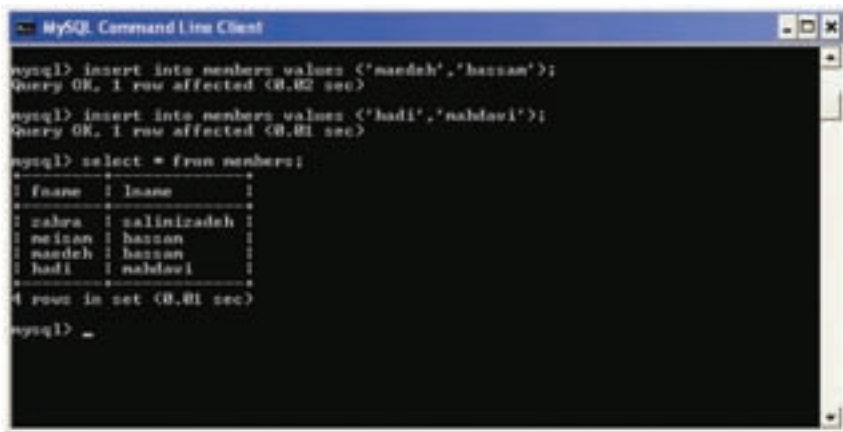
mysql> insert into members values ('meisam','hassan');
Query OK, 1 row affected (0.03 sec)

mysql> insert into members values ('masdeh','hassan');
Query OK, 1 row affected (0.02 sec)

mysql> insert into members values ('hadi','nabdavi');
Query OK, 1 row affected (0.01 sec)

mysql> _
```

شکل ۳-۶



```
mysql> insert into members values ('masdeh','hassan');
Query OK, 1 row affected (0.02 sec)

mysql> insert into members values ('hadi','nabdavi');
Query OK, 1 row affected (0.01 sec)

mysql> select * from members;
+-----+-----+
| fname | lname |
+-----+-----+
| zahra | salimizadeh |
| meisam | hassan |
| masdeh | hassan |
| hadi | nabdavi |
+-----+-----+
4 rows in set (0.01 sec)

mysql> _
```

شکل ۳-۷

مثال: مشخصات نام و نام خانوادگی را برای یک دانش آموز به نام Omid Safavi در جدول ۳-۱ وارد کنید.

شرح دستور وارد کردن نام و وضعیت جدید جدول:
کد:

```
insert into Student ( Name , Family )
values ( "Omid" , "Safavi" );
```


خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
10724113	Ehsan	Amiri	Electronic	Karaj	14
10254861	Zahra	Hoseini	Computer	Tehran	17
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19
	Omid	Safavi			

۷-۳ دستور update

این دستور برای تغییر اطلاعات موجود در یک فیلد جدول و جایگزینی آن با یک مقدار جدید به کار می‌رود.

شکل کلی این دستور به صورت زیر است:

نام جدول update

مقدار جدید = نام فیلد set

؛ مقدار قبلی = نام فیلد where

مثال: نام دانش‌آموز احمد رضایی را به علی تغییر دهید:

```
update Student
```

```
set Name = "Ali"
```

```
where Name = "Ahmad" and Family = "Rezaee";
```

• همچنین می‌توان برای تعیین رکوردی که می‌خواهیم تغییر دهیم، از مقدار یک فیلد دیگر به غیر از فیلدی که می‌خواهیم تغییر کند استفاده کنیم. در این حالت از فیلد مذکور، برای آدرس‌دهی رکورد مورد نظر استفاده می‌شود.

مثال: نام دانش‌آموز با شماره دانش‌آموزی ۴۱۲۵۲۲۱۴ را به علی در جدول ۱-۳ تغییر دهید:

• اعمال دستور تغییر و وضعیت جدید جدول:

کد:

```
update Student set Name = "Ali "
```

```
where StudentID = "41252214 ";
```

خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ali	Rezaee	Computer	Tehran	18
10724113	Ehsan	Amiri	Electronic	Karaj	14
10254861	Zahra	Hoseini	Computer	Tehran	17
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19

تغییر اطلاعات در بیش از یک فیلد در رکورد

در این حالت تمام مقادیر جدیدی که می‌خواهیم تغییر دهیم، جلوی واژه کلیدی Set به ترتیب نوشته و با کاما از هم جدا می‌کنیم. سپس به همان ترتیب، فیلدها با مقادیر قبلی را در جلوی واژه where می‌نویسیم. شکل کلی آن به صورت زیر است:

نام جدول update

مقدار = نام فیلد ۲ ، مقدار = نام فیلد ۱ set

where ... و مقدار قبلی = نام فیلد ۲ و مقدار قبلی = نام فیلد ۱

• همچنین می‌توان فقط از یک فیلد در قسمت شرط برای آدرس دهی رکورد مورد نظر استفاده کرد.

تمرین:

در جدول ۹-۲ از واحد کار دوم، مشخصات تولیدکننده P6 را تغییر دهید، طوری که وزن قطعه‌های تولید شده به ۲۵ و رنگ قطعه به زرد تبدیل شود.

مثال: در جدول ۱-۳ نام، نام خانوادگی و رشته تحصیلی دانش‌آموز به شماره دانش‌آموزی ۴۱۲۵۲۲۱۴ را به ترتیب به Ali، Saeedi و Electronic تغییر دهید:

• اعمال دستور تغییر و وضعیت جدید جدول:

کد:

```
update Student
```

```
set Name = "Ali" , Family = "Saeedi" , Major = "Electronic"
```

```
where StudentID = 41252214;
```

خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ali	Saeedi	Electronic	Tehran	18
10724113	Ehsan	Amiri	Electronic	Karaj	14
10254861	Zahra	Hoseini	Computer	Tehran	17
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19

۸-۳ دستور delete

از این دستور برای حذف اطلاعات یک رکورد در جدول استفاده می‌شود.
 شکل کلی این دستور به شکل زیر است:

نام جدول delete from

مقدار فیلد از رکورد موردنظر = نام فیلد where

مثال: در جدول ۱-۳، رکورد اطلاعات دانش‌آموز به شماره دانش‌آموزی ۱۰۷۲۴۱۱۳ را حذف کنید.

اعمال دستور حذف و وضعیت جدید جدول:
 کد:

deletefromStudent

where StudentID = "10724113";

خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
10254861	Zahra	Hoseini	Computer	Tehran	17
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19

حذف تمام رکوردهای یک جدول

این امکان وجود دارد که بدون حذف یک جدول، تمام رکوردهای درون آن را پاک کرد. این به منزله خالی کردن جدول از اطلاعات است، یعنی ساختار، اندیس‌ها و ویژگی‌های جدول دست نخورده باقی مانده و فقط مقادیر درون آن از بین می‌رود.

شکل کلی این دستور به صورت زیر است:

delete from جدول ;

یا

delete * from جدول ;

مثال: حذف تمام رکوردهای جدول Student

delete from Student;

۳-۹ دستور order by

اطلاعات خروجی در دستور Select به طور کلی بی‌نظم است و مقادیر خروجی در ستون‌های جدول برحسب مقدار هیچ ستونی مرتب نیستند. این دستور برای مرتب کردن اطلاعات جدول بر اساس مقادیر یک یا چند ستون برحسب شاخص‌هایی مثل ترتیب حروف الفبا، بزرگ‌تر یا کوچک‌تر بودن اعداد و ... استفاده می‌شود.

شکل کلی این دستور به صورت زیر است:

نام جدول مورد نظر from نام فیلدهای مورد نظر select

نام فیلد مورد نظر order by

فیلدهای معرفی شده در دستور select از جدول مذکور نمایش داده شده و اطلاعات بر حسب ستون مقابل دستور order by مرتب می‌شوند.

عبارت ASC: به کار بردن این دستور پس از دستور order by باعث مرتب شدن اطلاعات درون جدول از کم به زیاد می‌شود که البته انتخاب پیش‌فرض SQL است و نیازی به درج آن نیست.

عبارت DESC: به کار بردن این واژه پس از دستور order by باعث مرتب شدن اطلاعات جدول از زیاد به کم در اعداد و در حروف از Z تا A می‌شود.

مثال: از جدول Student، مشخصات نام و نام خانوادگی دانش‌آموزان رشته Electronic را برحسب حروف الفبا نمایش دهید.
کد:

```
select Name , Family from Student  
where Major ="Electronic"  
order by Family ;
```

خروجی

Name	Family
Sahar	Ahmadi
Ehsan	Amiri
Hesam	Razavi

نکته همچنین می‌توان یک جدول را بر حسب بیش از یک فیلد مرتب کرد، برای این منظور پس از دستور Order By نام تمام فیلدهای مورد نظر را نوشته و آن‌ها را با کاما از هم جدا می‌کنیم. اولویت مرتب‌سازی با فیلدهایی است که نام آن‌ها اول ذکر شده باشد.

مثال: جدول ۱-۳ را بر حسب فیلد نام خانوادگی (بر حسب حروف از A تا Z) و بر حسب نمره (از کم به زیاد) مرتب کنید:

نکته در جدول خروجی مثال فوق به ارتباط مقادیر ستون‌های Family و Grade توجه کنید. با وجود این که نمره درسی دانش‌آموز زهرا حسینی از احمد رضایی کمتر است، ولی به دلیل این که اولویت مرتب‌سازی با نام خانوادگی است، نام حسینی بالاتر از رضایی قرار گرفته است.

کد:

```
select Name , Family , Grade From Student  
order by Family ASC , Grade DESC ;
```

خروجی

Name	Family	Grade
Sahar	Ahmadi	16
Ehsan	Amiri	14
Zahra	Hoseini	17
Hesam	Razavi	19
Ahmad	Rezaee	18

نکته می‌توان در دستور order by به جای ذکر نام ستون‌ها از یک عدد به جای آن استفاده کرد. عدد استفاده شده، شماره قرارگیری ستون مورد نظر در دستور Select است. برای مثال اولین ستون دستور Select عدد ۱ و دومین ستون عدد ۲ و ... را خواهند داشت. بازنویسی دستور بالا با استفاده از این روش به صورت زیر خواهد بود:

کد:

```
select Name , Family , Grade from Student  
order by 2 ASC , 3DESC;
```

۱۰-۳ دستور and و or

از and و or برای ترکیب شرطها در دستور where استفاده می‌شود. گاهی اوقات اطلاعات موردنیاز ما در دستور select دارای چندین شرط مختلف است. به طور مثال ما مشخصات دانش‌آموزان ترم دوم رشته Computer که معدل آن‌ها بیش از ۱۵ است را می‌خواهیم. در این حالت باید هر یک از شرطهای مسئله را به تنهایی تعریف کرده و سپس آن‌ها را با هم ترکیب کنیم. برنامه به ارزیابی هر یک از شرطها پرداخته، سپس نتایج حاصله را با هم ترکیب کرده و خروجی را تعیین می‌کند. عملگر and برای اجرای دستور نیاز دارد تا تمام شرطهای تعیین شده برای آن، درست باشد. در حالی که عملگر or فقط نیاز دارد که حداقل یکی از شرطهای آن درست باشد. در جداول زیر چگونگی برقراری نتیجه نهایی در هنگام استفاده از این دو عملگر را مشاهده می‌کنیم:

and		
شرط ۱	شرط ۲	نتیجه
درست	درست	درست
درست	غلط	غلط
غلط	درست	غلط
غلط	غلط	غلط

or		
شرط ۱	شرط ۲	نتیجه
درست	درست	درست
درست	غلط	درست
غلط	درست	درست
غلط	غلط	غلط

چگونگی حالت‌های فوق را در مثال‌های زیر بررسی می‌کنیم:
مثال ۱: مشخصات دانش‌آموزانی را از جدول ۱-۳ (Student) ارائه دهید که نمره آن‌ها بیش از ۱۵ و رشته تحصیلی آن‌ها Electronic باشد:
 کد:

```
select * from Student
where Major ='Electronic' and Grade > 15 ;
```

خروجی

StudentID	Name	Family	Major	Area	Grade
27365187	Sahar	Ahmadi	Electronic	Bam	16
35654415	Hesam	Razavi	Electronic	Tehran	19

مثال ۲: مشخصات دانش‌آموزانی را از جدول Student ارائه دهید که نمره آن‌ها بیش از ۱۷ بوده یا رشته تحصیلی آن‌ها Computer باشد:
 کد:

```
select * from Student
where Major = 'Computer' OR Grade > 17 ;
```

خروجی

StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
35654415	Hesam	Razavi	Electronic	Tehran	19

نکته همچنین می‌توان با استفاده از عملگر پراتنز چندین شرط مختلف را به صورت دسته‌های مجزا تعریف کرده و سپس همه آن‌ها را با هم ترکیب کرد. در این حالت شرط‌های هر گروه ابتدا داخل پراتنز با هم ترکیب شده، سپس نتیجه حاصله به عنوان نتیجه نهایی آن گروه شرط به دستور Where ارسال می‌شود و در نهایت نتایج تمام گروه‌ها با هم ترکیب می‌شوند.

مثال: از جدول Student مشخصات دانش‌آموزانی را ارائه دهید که نمره آن‌ها ۱۸ و رشته تحصیلی آن‌ها Computer بوده یا نمره آن‌ها ۱۶ و رشته تحصیلی آن‌ها Electronic باشد:
 کد:

```
select * from Student
where ( ( Grade = 18 and Major = 'Computer' ) or ( Garde = 16 and Major = 'Electronic' ) ) " ;
```

خروجی

StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
27365187	Sahar	Ahmadi	Electronic	Bam	16

۱۱-۳ دستور in

این دستور برای مشاهده اطلاعات رکوردهایی از جدول به کار می‌رود که شما مقدار دقیق حداقل یکی از فیلدهای آن را می‌دانید.

شکل کلی این دستور به صورت زیر است:

نام جدول from نام فیلدهایی که می‌خواهید نمایش داده شود select
 ; (مقدار ۲ و مقدار ۱) in نام فیلد where

مثال: از جدول Student مشخصات دانش‌آموزانی را ارائه دهید که نام خانوادگی آنها

احمدی یا رضایی است:

کد:

```
select * from Student
```

```
where Family in ("Ahmadi", "Rezaee");
```

خروجی

Student Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
27365187	Sahar	Ahmadi	Electronic	Bam	16

۱۲-۳ دستور between ... and

این دستور برای انتخاب اطلاعات در یک محدوده خاص، در بین دو مقدار تعیین شده استفاده می‌شود. این مقادیر می‌تواند از نوع عددی، متن یا تاریخ باشد. (باید نوع مدنظر با نوع داده‌ای فیلدهای استفاده شده در دستور یکسان باشد).

شکل کلی این دستور به شکل زیر است:

نام جدول from نام فیلدهای مورد نظر برای نمایش select
نام فیلد مورد نظر where
مقدار ۲ and مقدار ۱ between

نکته در مورد داده‌های متنی، ترتیب قرارگیری حروف الفبا مد نظر برنامه است.

مثال: از جدول Student مشخصات دانش‌آموزانی را ارائه دهید که نام خانوادگی آنها بین دو مقدار احمدی و حسینی باشد. در ضمن خروجی بر حسب نام خانوادگی به ترتیب حروف الفبا مرتب باشد:
کد:

```
select * from Student  
where Family between 'Ahmadi' and 'Hoseini'  
order by family;
```

خروجی

StudentID	Name	Family	Major	Area	Grade
27365187	Sahar	Ahmadi	Electronic	Bam	16
10724113	Ehsan	Amiri	Electronic	Karaj	14
10254861	Zahra	Hoseini	Computer	Tehran	17

نکته قرار گرفتن رکوردهای ۲ مقدار تعیین شده در خروجی دستور select، به نوع برنامه پایگاه داده و مدیریت آن بستگی دارد. در برخی از پایگاه‌های داده رکوردهای هر دو مقدار ابتدا و انتها نمایش داده می‌شود و در برخی هم فقط یکی از آنها نمایش داده می‌شود.
برای نمایش اطلاعات خارج از محدوده تعیین شده از یک عملگر not قبل از دستور between ... and استفاده می‌شود.

مثال: از جدول ۱-۳ مشخصات دانش‌آموزانی را ارائه دهید که نمره آنها خارج از محدوده ۱۲ تا ۱۵ باشد. در ضمن مقادیر خروجی بر حسب نام خانوادگی به ترتیب حروف الفبا مرتب باشد:

کد:

```
select * from Student
where Grade not between 12 and 15
order by Family;
```

خروجی

StudentID	Name	Family	Major	Area	Grade
27365187	Sahar	Ahmadi	Electronic	Bam	16
10254861	Zahra	Hoseini	Computer	Tehran	17
35654415	Hesam	Razavi	Electronic	Tehran	19
41252214	Ahmad	Rezaee	Computer	Tehran	18

۱۳-۳ ویژگی alias

همانطور که در تمام مثال‌های ذکر شده از دستور select مشاهده کردید، نام یا عنوانی که در بالای هر ستون در خروجی نمایش داده می‌شود، همان نام فیلد مربوط به داده در جدول اصلی پایگاه داده است. گاهی اوقات تمایل بر این است که این نام را تغییر دهیم یا در مواردی که در بخش توابع SQL خواهیم دید، یک ستون جدید ایجاد می‌کنیم که باید نامی برای آن تعیین شود. از ویژگی alias برای در نظر گرفتن یک نام مستعار و موقت برای قرار گرفتن در بخش عنوان هر ستون در خروجی دستور select استفاده می‌شود. شکل کلی این دستور به صورت زیر است:

نام جدول from (alias) عنوان as نام فیلد select

مثال: از جدول Student نام و نام خانوادگی دانش‌آموزان رشته Electronic را بر حسب حروف الفبا در ۲ ستون تحت عنوان‌های "نام" و "نام خانوادگی" نمایش دهید:
 کد:

```
select Name as نام , Family as نام خانوادگی from Student
where Major = 'Electronic'
order by Family ;
```

خروجی

نام	نام خانوادگی
Sahar	Ahmadi
Ehsan	Amiri
Hesam	Razavi



نکته می‌توان روی مقادیر ستون‌هایی که داده عددی دارند، عملیات ریاضی (مثل ضرب، تقسیم و ...) انجام داده و سپس نتایج محاسبات را در یک ستون جدید با عنوان دلخواه تعیین شده توسط ویژگی `alias` در خروجی دستور `select` نمایش داد. در این حالت ستون یا ستون‌ها به جدول خروجی اضافه می‌شود.

در این ارتباط مثالی را ذکر می‌کنیم:

مثال: فرض کنید جدولی داریم به نام `Books` که در آن مشخصات چند کتاب مختلف قرار دارد. در این جدول قیمت کتاب‌ها به دلار است و ما می‌خواهیم قیمت کتاب‌ها را در یک ستون جدید به تومان مشاهده کنیم پس به این روش عمل می‌کنیم:

جدول ۳-۴

Books Table			
Price	Publisher	Name	no
8	Microsoft	Learning Access	1
6	DeveloperStudio	Java Script Samples	2
7	DeveloperStudio	HTML Reference	3

کد

`select no , Name , Publisher , Price , Price * 840 as قیمت به تومان from Books ;`

خروجی

Books Table				
no	Name	Publisher	Price \$	قیمت به تومان
1	Learning Access	Microsoft	8	6720
2	Java Script Samples	DeveloperStudio	6	5040
3	HTML Reference	DeveloperStudio	7	5880

۱۴-۳ پیوند جدول‌ها

تا این قسمت تمام مثال‌ها و مسئله‌هایی که در `SQL` به آن‌ها پاسخ دادیم، مسئله‌هایی بودند که در آن‌ها اطلاعات ما فقط از یک جدول استخراج می‌شد. اما در برنامه‌نویسی واقعی پایگاه داده‌ها، ما مجبور هستیم که اطلاعات خود را از بیش از یک جدول استخراج کنیم.

در این حالت ابتدا باید جدول‌هایی که می‌خواهیم اطلاعات را از آن‌ها استخراج کنیم، با هم پیوند دهیم. هدف از ایجاد این ارتباط تلفیق اطلاعات در جدول‌ها و چاپ اطلاعات مورد نظر در خروجی است.

برای پیوند دادن جدول‌ها به عنوان یادآوری این نکات را در نظر داشته باشید:
کلید اصلی: فیلد کلید اصلی در یک جدول، فیلدی است که شرایط زیر را داشته باشد: مقدار آن برای هر نمونه رکورد (سطر) منحصر به فرد و غیرتکراری باشد. به عبارت دیگر هیچ ۲ رکوردی در یک جدول در این فیلد مقدار یکسان نداشته باشند. کلید اصلی وجه تمایز ۲ نمونه رکورد مختلف در یک جدول است. طول مقادیر آن در حد امکان کوتاه باشد.

نکته یک جدول می‌تواند بیش از یک کلید اصلی داشته باشد.

مثال: فیلد شماره دانش‌آموزی در جدول Student، کلید اصلی است. هیچ دو دانش‌آموزی نمی‌توانند دارای شماره دانش‌آموزی یکسان باشند.

کلید خارجی: کلید خارجی، فیلدی است که در یک جدول کلید اصلی و در جدول دیگر به تنهایی کلید اصلی نباشد. از کلید خارجی برای ارتباط یک به چند ۲ جدول با هم استفاده می‌شود.

شرط ارتباط دو جدول

برای ارتباط بین جدول‌ها باید شرط‌های زیر برقرار باشد. باید قبل از طراحی پایگاه داده و جدول‌های آن موارد زیر را جهت ارتباط جدول‌های مورد نظر رعایت کرد:

- وجود فیلد مشترک دقیقاً از یک نوع و یک اندازه.
- فیلد مشترک در یکی از جدول‌ها کلید اصلی و در جدول دیگر کلید خارجی باشد.

نکته لزومی ندارد مقادیر موجود در جدول‌های مبدأ و مقصد یکسان باشد.

معرفی دو جدول دیگر

از این به بعد ما در مثال‌های خود از ۲ جدول دیگر به غیر از جدول Student، به

نام‌های Courses (درس‌ها) و Selection (انتخاب واحد) به شرح زیر استفاده می‌کنیم:

جدول ۳-۵

Courses Table			
CoType	Credit	CoTitle	CourseID
نوع درس	تعداد واحد	عنوان درس	(کلید اصلی) کد درس

جدول ۳-۶

Selection Table				
Grade	Year	Term	CourseID	StudentID
نمره	سال تحصیلی	ترم تحصیلی	کد درس (کلید اصلی خارجی)	شماره دانش آموزی (کلید اصلی خارجی)

نکته • در تمام مثال‌های قبلی، ما در دستور select فقط نام ستون‌ها را به تنهایی ذکر می‌کردیم، زیرا در آن زمان، اطلاعات ما فقط از یک جدول استخراج می‌شد. اما در هنگام پیوند دو جدول و استفاده از چند جدول در دستور select باید نام ستون را به همراه نام جدول مربوط به آن ذکر کرد. این کار ۲ دلیل اصلی دارد:

- باعث تمایز ستون‌های مشترک در جدول‌ها از یکدیگر می‌شود و مشخص می‌کند که هر ستون مربوط به کدام جدول است.
- باعث خوانایی و دقت بیشتر برنامه می‌شود.

شکل کلی این دستور به صورت زیر است:

نام ستون . نام جدول

مثال: انتخاب ستون StudentID از جدول Student:

Student.StudentID

۱-۱۴-۳ مثال‌های پیوند جدول‌ها

در این قسمت با ارائه چندین مثال، انواع حالت‌های مختلف پیوند جدول‌ها را بررسی می‌کنیم. از داده‌های موجود در جداول زیر برای مثال‌ها استفاده می‌کنیم:

توجه: جدول انتخاب واحد نشان‌دهنده این است که هر دانش‌آموز چه واحدهای درسی را در چه ترم و سالی و با چه نمره‌ای گذرانده است.

جدول ۳-۷

Courses Table			
CoType	Credit	CoTitle	CourseID
عملی	3	پایگاه داده	1011
عملی	3	سخت‌افزار	1012
نظری	2	زبان تخصصی	1013

جدول ۳-۸

Selection Table				
Grade	Year	Term	CourseID	StudentID
16	88 - 89	2	1011	41252214
14	88 - 89	2	1011	10724113
17	88 - 89	1	1012	41252214
11	88 - 89	1	1012	10724113
13	88 - 89	2	1013	10254861
8	87 - 88	2	1011	10254861
19	87 - 88	1	1012	27365187
16	87 - 88	1	1013	27365187
9	87 - 88	2	1011	35654415
17	87 - 88	2	1013	35654415

شکل کلی پیوند ۲ جدول برای استخراج اطلاعات به صورت زیر است:

نام رکورد یا رکوردهای مورد نظر برای نمایش select

نام جدول‌ها from

برابر قرار دادن فیلدهای مشترک ۲ جدول where

; بقیه شرط‌های مورد نظر and

در این حالت ابتدا در دستور select نام ستون‌هایی را که می‌خواهیم از ۲ جدول نمایش دهیم، تعیین می‌کنیم. سپس نام ۲ جدول را در مقابل دستور from نوشته و در اولین شرط دستور where، نام فیلد مشترک را از هر ۲ جدول نوشته و آن‌ها را برابر هم قرار می‌دهیم. این شرط، شرط برقراری پیوند و تلفیق اطلاعات ۲ جدول است. در ادامه هم می‌توان شرط‌های دیگری را برای استخراج اطلاعات تعیین کرد. در مثال‌های بعدی این مسئله را بررسی می‌کنیم:

مثال: نام و نام خانوادگی و سال ورود دانش آموزانی را ارائه دهید که در ترم ۱ سال تحصیلی ۸۸-۸۹، درسی را با کد ۱۰۱۲ انتخاب کرده‌اند:
 کد:

```
select Student.Name , Student.Family , Selection.Term , Selection.Year
from Student , Selection
where Student.StudentID = Selection.StudentID
and CourseID = 1012 and Term = 1 and Year = '88 - 89'
order by Student.Family;
```

خروجی

Name	Family	Term	Year
Ehsan	Amiri	1	88 - 89
Ahmad	Rezaee	1	88 - 89

مثال: نام دروسی را ارائه دهید که دانش آموز با شماره دانش آموزی ۱۰۲۵۴۸۶۱ انتخاب کرده است:
 کد:

```
select Courses.CourseID , Courses.Co Title
from Courses , selection
where Courses.CourseID = Selection.CourseID
and Selection.StudentID = 10254861 ;
```

خروجی

CourseID	CourseTitle
1011	پایگاه داده
1013	زبان تخصصی

مثال: نام و نام خانوادگی دانش آموزانی را ارائه دهید که درس با کد ۱۰۱۳ در سال تحصیلی ۸۷-۸۸ را با نمره بالاتر از ۱۵ گذرانده‌اند:
 کد:

```
select Student.Name , Student.Family
from Student , Selection
where Student.Studentid = Selection.Studentid
and Selection.Courseid = '1013' and Year = '87 - 88' and Grade > 15 ;
```

خروجی

Name	Family
Sahar	Ahamdi
Hesam	Razavi

۲-۱۴-۳ پیوند بیش از دو جدول به یکدیگر

گاهی اوقات لازم است که اطلاعات مورد نیاز ما از ۳ جدول یا بیشتر استخراج شود. در این حالت باید تمام جدول‌ها را به هم پیوند دهیم، به این صورت که معمولاً از یک جدول سوم برای پیوند ۲ جدول دیگر استفاده می‌شود و دو به دو جدول‌هایی که با هم فیلد مشترک دارند را با ذکر شرط پیوند در دستور where به هم پیوند می‌دهیم. سپس بقیه شروط دلخواه را نیز ذکر می‌کنیم.

شکل کلی این حالت به این صورت است:

```
نام ستون‌های مورد نظر از جدول‌ها select  
نام تمام جدول‌ها from  
برابر قرار دادن فیلد مشترک جدول‌های ۱ و ۲ where  
برابر قرار دادن فیلدهای مشترک جدول‌های ۲ و ۳ and  
and ... ;
```

مثال: نام و نام خانوادگی دانش‌آموزانی را ارائه دهید که حداقل یک درس از نوع نظری را انتخاب کرده باشند:

کد:

```
select Students.Name , Students.Family , Courses.CoTitle , Courses.CoType "  
from Students , Courses , Selections  
where Student.StudentID = Selection.StudentID  
and Courses.CourseID = Selection.CourseID ;  
and Courses.CoType = 'نظری' ;
```

خروجی

Name	Family	CoTitle	CoType
Zahra	Hoseini	زبان تخصصی	نظری
Sahar	Ahamadi	زبان تخصصی	نظری
Hesam	Razavi	زبان تخصصی	نظری

با دقت در اطلاعات جدول‌های اصلی متوجه درست بودن نتایج خروجی خواهید شد.

۱۵-۳ دستور join

از دستور join برای پیوند دو یا چند جدول در پایگاه داده‌ها استفاده می‌شود. قبل از بیان دستور Join به جدول‌های زیر دقت کنید:

a1	b1
a2	b1
a3	b2

a

b1	c1
b2	c2
b3	c3

b

می‌خواهیم عمل پیوند دو جدول a و b را طوری انجام دهیم که یک جدول با سه ستون ایجاد شود، در این صورت جدول حاصل به صورت زیر خواهد بود:

a1	b1	c1
a2	b1	c1
a3	b2	c2

برای ایجاد جدول سوم، سطرهای جدول اول با جدول دوم سنجیده می‌شوند، به عنوان مثال در سطر اول، b1 و a1 با هم متناظرند، و در جدول دوم عنصر b1 با عنصر c1 متناظر است. در نتیجه عنصر c1 به عنوان عنصر متناظر b1 و a1 در جدول سوم است. (a1 و b1، c1).

در سطر سوم از جدول اول نیز b2 و a3 با هم‌دیگر متناظرند، در جدول دوم، عنصر b2 با عنصر c2 متناظر است. در نتیجه عنصر c2 به عنوان عنصر متناظر b2 و a3 در جدول سوم است. (a3 و b2، c2).

عمل join (الحاق) رابطه‌ای را برمی‌گرداند که شامل تمام ترکیب‌های حاصل از دو رابطه اولیه است. برای پیوند جدول‌ها با استفاده از دستور join، از روش کلی زیر استفاده می‌شود:

نام ستون‌های موردنظر برای نمایش از هر دو جدول select

نام جدول اول from

نام جدول دوم inner join

برابر قرار دادن فیلدهای مشترک دو جدول on

سایر شرط‌ها where

مثال: می‌خواهیم در جدول‌های Selection و Student، نام و نام‌خانوادگی، ترم و سال

تحصیلی دانش‌آموزانی را استخراج کنیم که در ترم ۱ سال تحصیلی ۸۸-۸۹، درسی با کد ۱۰۱۲ را انتخاب کرده‌اند:
کد:

```
select Student.Name , Student.Family , Seleciton.Term , Selection.Year
from Student
inner join Seleciton
on Student.StudentID = Selection.StudentID
where Selection.CourseID = 1012 and Term = 1 and Year = ' 88 - 89 ' ;
```

خروجی

Name	Family	Term	Year
Ehsan	Amiri	1	88 - 89
Ahmad	Rezaee	1	88 - 89

مثال: نام و شماره درس‌هایی را ارائه دهید که دانش‌آموز با شماره دانش‌آموزی ۱۰۲۵۴۸۶۱ انتخاب کرده است:
کد:

```
select Courses.CourseID , Courses.Co Title
from Courses
inner join Selection
on Courses.CourseID = Selection.CourseID
where Selection.StudentID = 10254861 ;
```

خروجی

CourseID	CourseTitle
1011	پایگاه داده
1013	زبان تخصصی

۳-۱۶ Select های تودرتو

می‌توان درون یک دستور select، دستور select دیگری را تعریف کرد. در این حالت ابتدا دستور Select درونی اجرا شده و نتایج مورد نظر خود را که سطرهایی از جدول مربوط به آن هستند استخراج کرده و آن را به select بیرونی انتقال می‌دهد. سپس select

بیرونی نیز اجرا شده و نتایج استخراجی خود را با استفاده از فیلد مشترک بین ۲ جدول با سطرهای select درونی پیوند زده و سطرهای مطابق با شرطهای تعیین شده را نمایش می‌دهد.

نکته به تعداد دستورات select تودرتو، سطح آن می‌گویند.

به علت پیچیدگی selectهای ۳ سطحی، ۴ سطحی و ... در این واحد کار فقط دستور select دو سطحی را بررسی می‌کنیم.

۱-۱۶-۳ selectهای ۲ سطحی

در این حالت یک دستور select درون دستور select دیگری تعریف می‌شود. نتایج خروجی فقط می‌تواند شامل سطرهایی از جدول Select بیرونی باشد. شکل کلی select دو سطحی به صورت زیر است:

نام ستونهای مورد نظر برای نمایش از جدول بیرونی select

نام جدول ۱ from

نام فیلد مشترک دو جدول (Select) in نام فیلد مشترک دو جدول where

نام جدول ۲ From

(... شرطها Where);

مثالهای قسمت‌های قبل پیوند جدول‌ها را با استفاده از Selectهای تودرتو باز نویسی می‌کنیم

مثال: نام و نام خانوادگی دانش‌آموزانی را ارائه دهید که در ترم ۱ سال تحصیلی ۸۹-

۸۸، درس با کد ۱۰۱۲ را انتخاب کرده‌اند:

کد:

```
select Name , Family
```

```
from Student
```

```
where StudentID in (select StudentID
```

```
from Selection
```

```
where CourseID = 1012 and Term = 1 and Year = ' 88 - 89 ' );
```

خروجی

Name	Family
Ehsan	Amiri
Ahmad	Rezaee

نکته تفاوت این حالت با مثال‌های قبلی در این است که ستون‌های نمایش داده شده فقط از یک جدول (جدول select بیرونی) نمایش داده می‌شوند.

مثال: شماره و نام درس‌هایی را ارائه دهید که دانش آموز با شماره دانش‌آموزی ۱۰۲۵۴۸۶۱ انتخاب کرده است:
کد:

```
select CourseID , Co Title
from Courses
where CourseID in ( Select CourseID
                    from Selection
                    where StudentID = 10254861 ) ;
```

خروجی

CourseID	CourseTitle
1011	پایگاه داده
1013	زبان تخصصی

۱۷-۳ دستور union

از دستور Union برای ترکیب و ادغام دو یا چند ستون مختلف از ۲ یا چند جدول استفاده می‌شود. دستور union نتیجه را در یک ستون مشترک نمایش می‌دهد. در این دستور، نوع داده ستون‌های انتخاب شده برای ادغام باید یکسان باشند. دستور union در هنگام ترکیب فیلدها، در صورت برخورد با مقادیر تکراری، آن‌ها را حذف کرده و از هر مقدار یک نمونه را نمایش می‌دهد. برای مشاهده تمام مقادیر، حتی مقادیر تکراری باید از دستور union all استفاده کرد. شکل کلی استفاده از این دستور به صورت زیر است:

(نام فیلد ۱ انتخابی) SQL Statement 1

union

(نام فیلد ۲ انتخابی) SQL Statement 1 ;

مثال: در جدول ۱-۳، شماره دانش آموزی دانش آموزانی را نمایش دهید که یا دانش آموز رشته Electronic بوده و درس سخت افزار با کد ۱۰۱۲ را انتخاب کرده اند یا در درس زبان تخصصی با کد ۱۰۱۳ ثبت نام کرده اند:

```
select StudentID from Student
where Major = 'Electronic' and CourseID = 1012
union
select StudentID from Selection ;
where CourseID = 1013 ;
```

خروجی

StudentID
10254861
10724113
27365187
35654415

نکته می توان در دستور union بیش از یک ستون مجزا را تعریف کرد. فقط باید تعداد ستون ها و ترتیب نوع داده ای ستون های تعریف شده یکسان باشند. برنامه مقادیر ستون ها را به ترتیب با هم پیوند می دهد.

مثال: شماره دانش آموزی، نام و نام خانوادگی دانش آموزانی را نمایش دهید که در رشته Electronic تحصیل کرده یا حداقل یک درس از نوع نظری را انتخاب کرده باشند:

```
select StudentID , Name , Family
from Student
where Major = 'Electronic'
union
select StudentID , Name , Family
from Student , Selection , Courses
where Student.StudentID = Selection.StudentID
and Selection.CourseID = Courses.CourseID "
and CoType = 'نظری' ;
```

خروجی

StudentID	Name	Family
10254861	Zahra	Hoseini
107241113	Ehsan	Amiri
27365187	Sahar	Ahmadi
35654415	Hesam	Razavi

۱۸-۳ دستور drop

از این دستور برای حذف یک جزء در محیط پایگاه داده استفاده می‌شود.

۱۸-۳-۱ حذف اندیس (drop index)

از این دستور برای حذف یک اندیس موجود در یک فیلد جدول استفاده می‌شود. شکل کلی این دستور به شرح زیر است:

نام جدول on نام اندیس drop index
مثال: drop index Rowup on Student ;

۱۸-۳-۲ حذف جدول یا پایگاه داده (drop DataBase یا drop table)

از این دستور برای حذف یک جدول در پایگاه داده یا یک پایگاه داده به صورت کامل استفاده می‌شود.
دستور حذف جدول:

نام جدول drop table ;


دستور حذف پایگاه داده:

مثال drop table Student ;

نام پایگاه داده drop DataBase ;

مثال: اگر پایگاه داده‌ای به نام School داشته باشیم، دستور حذف آن به شکل زیر است:

drop DataBase School ;

نکته  برای حذف تمام اطلاعات موجود در یک جدول بدون حذف ستون‌های

آن از دستور زیر استفاده می‌شود:

نام جدول truncate table ;

truncate table School ; مثال

این دستور باعث خالی شدن جدول از اطلاعات می شود، بدون این که خود جدول و ساختارش تغییری کند.

۱۹-۳ دستور alter

از دستور alter برای اضافه کردن یا حذف کامل یک ستون از یک جدول موجود در پایگاه داده استفاده می شود.

۱-۱۹-۳ اضافه کردن یک ستون

برای اضافه کردن یک ستون (فیلد) جدید به یک جدول موجود از دستور کلی زیر استفاده می شود:

نام جدول alter table

نوع داده ستون نام ستون add

مثال: اضافه کردن یک ستون جدید به نام Age به جدول Student:

فیلدهای جدول Student قبل از اجرای دستور

Student Table					
StudentID	Name	Family	Major	Area	Grade

کد:

```
alter table Student
```

```
add Age Int ;
```

فیلدهای جدول Student پس از اجرای دستور

Student Table						
StudentID	Name	Family	Major	Area	Grade	Age

۲-۱۹-۳ حذف ستون

برای حذف یک ستون از یک جدول موجود به روش زیر عمل می شود:

نام جدول alter table

نام ستون drop column

مثال: حذف فیلد Age از جدول Students :

جدول Student قبل از اجرای دستور

Student Table						
StudentID	Name	Family	Major	Area	Grade	Age

کد:

alter table Student
 drop column Age ;

جدول Student پس از اجرای دستور

Student Table					
StudentID	Name	Family	Major	Area	Grade

۲۰-۳ توابع درون ساخته SQL

زبان SQL دارای یکسری توابع درون ساخته (آماده) برای محاسبات روی داده‌های جداول و ارائه خروجی مناسب برای اهداف خاص است. در هنگام استفاده از توابع SQL باید به نکات زیر دقت کرد: باید نوع داده هر ستون (فیلد) جدول متناسب با عملکرد تابع مورد استفاده باشد. برای مثال توابعی که اعمال ریاضی انجام می‌دهند، باید به فیلدهایی از نوع عددی اعمال شوند. می‌توان نتایج عملکرد هر تابع را با استفاده از ویژگی alias که قبلاً در مورد آن صحبت کردیم، به صورت یک ستون جدید با عنوان دلخواه نمایش داد. برای مثال می‌توان با استفاده از تابع avg، میانگین نمرات هر دانش‌آموز را محاسبه کرده و آن را در یک ستون جدید به نام average نمایش داد. این مورد را در مثال‌های ذکر شده بررسی می‌کنیم. برنامه، نتایج خروجی هر تابع را به صورت یک ستون جدید نمایش می‌دهد. شکل کلی استفاده از توابع درون ساخته به صورت زیر است:

نام جدول from (نام فیلد) *function select

* function = نام تابع ;

مثال: محاسبه و نمایش معدل دانش‌آموزان با استفاده از تابع Avg و نمایش آن تحت

ستون جدیدی به نام دلخواه:

average :

select avg (Grade) as Average from Student ;

نکته در هنگام استفاده از توابع SQL، چنانچه شرطها برای دستور select تعریف شده باشد، تابع فقط مقادیر فیلدهایی از هر ستون را در محاسبه دخالت می دهد که با شروط ارائه شده سازگاری داشته باشند.

برای مثال وقتی می گوئیم که معدل دانش آموزان رشته Computer را در درس پایگاه داده محاسبه کنید، برنامه فقط میانگین نمرات درس پایگاه داده دانش آموزان را محاسبه می کند و به نمرات سایر دروس کاری ندارد.

نکته چنانچه توسط ویژگی alias، عنوانی برای ستون جدید ایجاد شده تعیین نکنید، خود برنامه به صورت خودکار یک نام پیش فرض به آن اختصاص می دهد.

۱-۲۰-۳ تابع avg

تابع avg برای محاسبه میانگین اعداد موجود در یک فیلد به کار می رود. این تابع اعداد را با هم جمع کرده و آنها را به طور اتوماتیک بر تعدادشان تقسیم می کند و نتیجه را به عنوان خروجی برمی گرداند.

نکته این تابع باید با فیلدهایی که داده آنها از نوع عددی است، اعمال شود.

مثال: معدل نمرات دانش آموزان را از جدول Student محاسبه کرده و در ستونی به نام معدل نمایش دهید:

کد:

```
select avg ( Students.Grade ) as معدل from Student
```

خروجی

معدل
16.8

مثال: معدل نمرات دانش آموزان رشته Computer را در درس پایگاه داده محاسبه کرده و در ستونی به نام معدل نمایش دهید:

کد:

```
select avg ( Selection.Grade ) as معدل
```

```
from Students,Selection,Courses
where Students.StudentID = Selection.StudentID
and Selection.CourseID = Courses.CourseID
and Major = 'Computer'
and CoTitle = 'پایگاه داده' ;
```

خروجی

معدل
11.5

۲-۲۰-۳ تابع sum

تابع sum، مجموع مقادیر اعداد در یک فیلد را محاسبه کرده و به عنوان خروجی برمی گرداند.

نکته این تابع باید با فیلدهایی که داده آنها از نوع عددی است، به کار رود.

مثال: از جدول انتخاب واحد (Selection) مجموع نمرات دانش آموزان را در درس پایگاه داده با کد ۱۰۱۱ محاسبه کنید:
کد:

```
select Sum ( Selection.Grade ) as مجموع نمرات پایگاه داده
from Selection,Courses
and Selection.CourseID = Courses.CourseID
and Selection.CourseID = 1011 ;
```

خروجی

مجموع نمرات پایگاه داده
47

مثال: از جدول انتخاب واحد، مجموع نمرات دانش آموز حسام رضوی را محاسبه کنید:
کد:

```
select Sum ( Selection.Grade ) as مجموع نمرات حسام رضوی
from Student,Selection
where Student.StudentID = Selection.StudentID
```

```
and Name = ' Hesam '
```

```
and Family = ' Razavi ' ;
```

خروجی

مجموع نمرات حسام رضوی
26

۳-۲۰-۳ تابع min

این تابع، کمترین مقدار موجود در بین مقادیر فیلدهای یک ستون را برمی‌گرداند.

نکته این تابع با ستون‌هایی که داده آن‌ها از نوع عددی یا متنی باشد، می‌تواند به کار رود. در فیلدها با مقادیر عددی تابع کوچک‌ترین عدد و در فیلدها با مقادیر متنی، تابع کلمه‌ای را که به ترتیب حروف الفبا از آ تا ی در فارسی و A تا Z در انگلیسی در بالاترین رده باشد برمی‌گرداند.

مثال: کوچک‌ترین نام خانوادگی از نظر حروف الفبایی را از جدول Student نمایش

دهید:

کد:

```
select min (Family) as Family from Student ;
```

خروجی

Family
Ahmadi

مثال: کمترین نمره دانش‌آموزان را از جدول Selection نمایش دهید:

کد:

```
select min ( Grade ) as کمترین نمره from Selection ;
```

خروجی

کمترین نمره
8

۳-۲۰-۴ تابع max

این تابع، بیشترین مقدار موجود در بین مقادیر فیلدهای یک ستون را برمی‌گرداند.

نکته این تابع با ستون‌هایی که داده آن‌ها از نوع عددی یا متنی باشد، می‌تواند به کار رود. در فیلدها با مقادیر عددی تابع بزرگ‌ترین عدد و در فیلدها با مقادیر متنی، تابع کلمه‌ای را که به ترتیب حروف الفبا از آ تا ی در فارسی و A تا Z در انگلیسی در آخرین رده باشد برمی‌گرداند.

مثال: آخرین نام خانوادگی دانش‌آموزان را از جدول Student نمایش دهید:

کد:

select max (Family) as Family from Student;

خروجی

Family

Rezaee

مثال: بیشترین نمره دانش‌آموزان را از جدول Selection نمایش دهید:

کد:

select max (Grade) as نمره بیشترین from Selection ;

خروجی

بیشترین نمره

19

۵-۲۰-۳ تابع first

این تابع مقدار اولین رکورد را در یک فیلد برمی‌گرداند. ترتیب در اینجا، همان ترتیب قرارگیری رکوردها در جدول است.

۶-۲۰-۳ تابع last

این تابع مقدار آخرین رکورد را در یک فیلد برمی‌گرداند. ترتیب در اینجا، همان ترتیب قرارگیری رکوردها در جدول است.

۷-۲۰-۳ تابع count

این تابع تعداد سطرهای موجود در یک فیلد را شمرده و مقدار آن را برمی‌گرداند. به این تابع، تابع سطر شمار هم می‌گویند.

مثال: این تابع فیلدهایی را که مقدار آن‌ها خالی یا تکراری باشد، نمی‌شمارد. برای شمارش تمام فیلدها (حتی تکراری) باید قبل از نام ستون در دستور count عبارت distinct ذکر شود.

مثال: تعداد درس‌هایی که دانش‌آموز با شماره دانش‌آموزی ۳۵۶۵۴۴۱۵ قبول شده را

ارائه دهید:

کد:

select count (CourseID) as تعداد دروس

from Selection

where StudentID = 35654415

and Grade > 10 ;

خروجی

تعداد دروس

1

مثال: تعداد دانش‌آموزانی که درس پایگاه داده را انتخاب کرده‌اند اعلام کنید:

کد:

select count (distinct Selection.CourseID) as تعداد دانش‌آموزان

from Selection , Courses

where Selection.CourseID = Courses.CourseID

and CoTitle = ' پایگاه داده ' ;

خروجی

تعداد دانش‌آموزان

4

۸-۲۰-۳ دستور group by

در هنگام استفاده از برخی توابع درون ساخته SQL که عمل محاسبه (مثل مجموع و میانگین) را روی داده‌ها انجام می‌دهند، این مشکل وجود دارد که این توابع قادر به جدا کردن و متمایز کردن اطلاعات موجود در دو ستون نسبت به هم نیستند و نتایج محاسبات را به صورت کلی برای همه آن‌ها در نظر می‌گیرند.

از دستور group by برای دسته‌بندی یک ستون بر حسب مقادیر مشابه فیلدهای یک ستون دیگر استفاده می‌شود.

شکل کلی استفاده از این دستور به صورت زیر است:

... و نام فیلد ۲ و نام فیلد ۱ select

from نام جدول

group by نام فیلد ;

مثال ۱: فرض کنید جدولی به صورت زیر برای نمرات ۲ دانش‌آموز داریم، می‌خواهیم مجموع نمرات هر یک از این دو دانش‌آموز را نمایش دهیم:

جدول ۳-۹

Class Table	
Grade	Name
18	Ali Ahmadi
17	Reza Hoseini
20	Ali Ahmadi
13	Reza Hoseini

کد:

select sum (Grade) as مجموع نمرات
 from Class ;

در صورت عدم استفاده از دستور group by تابع Sum خروجی زیر را خواهیم داشت:
 خروجی

مجموع نمرات
68

• مسلماً خروجی فوق مورد نظر نیست. می توان با استفاده از دستور group by، مجموع نمرات را برای دانش آموزان بر حسب ستون Name به صورت جدا محاسبه کرد و به این صورت نمایش داد:
 کد:

select Name ,sum (Grade) as مجموع نمرات
 from Class
 group by Name ;

خروجی	
Name	مجموع نمرات
Ali Ahmadi	38
Reza Hoseini	30

مثال ۲: از جدول Student و Selection مجموع نمرات هر دانش آموز را به همراه نام و نام خانوادگی که بر حسب نام خانوادگی مرتب شده باشد نمایش دهید:

کد:

select Name , Family , sum (Selection.Grade) as مجموع نمرات

from Student , Selection
 where Student.StudentID = Selection.StudentID
 group by Name , Family
 order by Family;

خروجی

Name	Family	مجموع نمرات
Sahar	Ahmadi	35
Ehsan	Amiri	25
Zahra	Hoseini	21
Hesam	Razavi	26
Ahmad	Rezaee	33

۹-۲۰-۳ دستور having

از این دستور برای افزودن شرط به توابع درون ساخته SQL استفاده می شود، زیرا از دستور where نمی توان برای کار با مقادیر خروجی توابع درون ساخته SQL استفاده کرد. به عبارت دیگر having برای اعمال شرط به ستون‌ها اعمال می شود و همان کاری را می کند که where در رکوردها انجام می دهد. با مثال‌های بعدی استفاده از دستور having را بررسی می کنیم:

نکته دستور having معمولاً با group by می آید.

مثال: از جدول Selection و Student مجموع نمرات و نام و نام خانوادگی دانش آموزانی را که مجموع نمرات آنها بیش از ۲۵ است نمایش دهید:
 کد:

select Name , Family , sum (Selection.Grade) as مجموع نمرات
 from Student , Selection
 where Student.StudentID = Selection.StudentID
 group by Name , Family
 having Sum (Selection.Grade) > 25
 order by Family;

خروجی

Name	Family	مجموع نمرات
Sahar	Ahmadi	۳۵
Ahmad	Rezaee	۳۳

مثال: از جدول Selection و Courses نام دروسی را ارائه دهید که ۴ بار توسط دانش‌آموزان انتخاب شده باشد:
کد:

```
select CoTitle , Count (Selection.CourseID) as تعداد انتخاب
from Selection , Courses
where Selection.CourseID = Courses.CourseID
group by CoTitle
having Count (Selection.Grade) = 4 ;
```

خروجی

CoTitle	تعداد انتخاب
پایگاه داده	4


select into دستور ۳-۲۰-۱۰

از دستور select into در موارد زیر استفاده می‌شود:

- ایجاد یک Backup (نسخه پشتیبان) از یک جدول
- ایجاد یک آرشیو از رکوردهای یک جدول
- قرار دادن برخی از رکوردها یا فیلدهای مورد نظر از یک جدول در یک جدول جدید
- ایجاد یک نسخه پشتیبان از کل یک پایگاه داده در یک پایگاه داده جدید

شکل کلی استفاده از این دستور به صورت زیر است:

```
select [نام یک پایگاه داده جدید in] نام جدول جدید into نام فیلدهای مورد نظر
from نام جدول مبدأ ;
```

نکته  ذکر نام یک پایگاه داده جدید در دستور select into اختیاری است. چنانچه نامی در این قسمت ذکر نشود، برنامه نسخه پشتیبان را در همان پایگاه داده موجود ایجاد می‌کند و چنانچه نامی ذکر شود، برنامه نسخه پشتیبان را در پایگاه داده ذکر شده ایجاد می‌کند.

مثال: ایجاد یک Backup از جدول Student در یک جدول جدید:

```
select * into Student_Backup
from Student ;
```


مثال: ایجاد یک Backup از جدول Student در یک جدول دیگر در یک پایگاه داده جدید به نام Backup.mdb:

```
select * into Student_Backup in «Backup.mdb»
```

```
from Student ;
```

توجه: در مثال‌های زیر از اطلاعات جداول Selection، Courses، و Student که در صفحات قبل معرفی کردیم، استفاده می‌کنیم.

نکته همچنین می‌توان فقط یک یا چند ستون یا رکورد را از یک جدول در یک جدول جدید وارد کرد. برای این کار باید نام آن فیلدها را پس از دستور select تعیین کرده و برای رکوردهای مورد نظر نیز شروط مورد نظر تعریف شود.

مثال: فیلدهای Name و Family را از جدول Student در یک جدول جدید به نام Names وارد کنید:
کد:

```
select Name , Family into Names
```

```
from Student ;
```

بعد از اجرای دستور فوق جدول جدیدی به نام Names با مشخصات بعد ایجاد خواهد شد.

خروجی

Name	Family
Ahmad	Rezaee
Ehsan	Amiri
Zahra	Hoseini
Sahar	Ahmadi
Hesam	Razavi

مثال: از جدول Student، مشخصات دانش‌آموزانی که نمره آن‌ها بیش از ۱۷ است را در جدول جدیدی به نام Highscores وارد کنید:
کد:

```
select * into Highscores
```

```
from Student
```

```
where Grade > 17 ;
```

بعد از اجرای دستور فوق جدول جدیدی به نام Highscores Table با مشخصات زیر ایجاد خواهد شد.

خروجی

Highscores Table					
StudentID	Name	Family	Major	Area	Grade
41252214	Ahmad	Rezaee	Computer	Tehran	18
35654415	Hesam	Razavi	Electronic	Tehran	19

مثال: مشخصات نام، نام خانوادگی، نام درس و نمره دانش‌آموزانی که در جدول انتخاب واحد، نمره آن‌ها بیش از ۱۵ بوده را در یک جدول جدید به نام Highgrades وارد کنید:
 کد:

```
select Name , Family , CoTitle , Selection.Grade Into Highgrades
from Student , Selection , Courses
where Student.StudentID = Selection.StudentID
and Selection.CourseID = Courses.CourseID
and Grade > 15 ;
```

بعد از اجرای دستور فوق جدول جدیدی به نام Highgrades Table با مشخصات زیر ایجاد خواهد شد.

خروجی

Highgrades Table			
Name	Family	CoTitle	Grade
Ahmad	Rezaee	پایگاه داده	16
Sahar	Ahmadi	مباحث ویژه	19
Ahmad	Rezaee	مباحث ویژه	17
Sahar	Ahmadi	زبان تخصصی	16
Hesam	Razavi	زبان تخصصی	17

واژه‌نامه

A verage	میانگین
Coun t	تعداد
First	اولین
Grou p	گروه
Havi ng	داشتن
Last	آخرین
Max imum	حداکثر
Mini mum	حداقل
Sum	حاصل جمع

خلاصه مطالب

- زبان SQL یک زبان ساخت یافته برای اتصال و ارتباط و کار با بانک‌های اطلاعاتی است.
- این زبانه با ارائه دستورات مختلف و کاربردی در تمام سیستم‌های مدیریت بانک‌های اطلاعاتی قابل دسترسی و استفاده است.
- برای اجرای دستورات SQL و مشاهده عملکرد آن‌ها، باید از یک سیستم مدیریت پایگاه داده‌ها استفاده شود.
- دستور create برای ایجاد یک جزء جدید در سیستم‌های پایگاه داده استفاده می‌شود.
- دستور select برای انتخاب و استخراج اطلاعات مورد نظر از یک یا چند جدول و مشاهده نتایج استفاده می‌شود.
- دستور insert برای وارد کردن و ایجاد یک رکورد جدید در جدول استفاده می‌شود.
- دستور update برای تغییر اطلاعات موجود در یک فیلد جدول استفاده می‌شود.
- دستور delete برای حذف اطلاعات یک رکورد در جدول استفاده می‌شود.

آزمون نظری

۱ - برای ایجاد یک پرسش و استخراج اطلاعات جدول از استفاده می شود.

الف-DML ب- رکورد ج- Query د- DDL

۲- کدام یک از گزینه های زیر در عملیات جستجو، باعث بالا رفتن سرعت جستجو می شود؟

الف- Query ب- Index ج- MySQL د- Create

۳ - دستور Create برای ایجاد کدام یک از گزینه های زیر به کار نمی رود؟

الف- پایگاه داده ب- جدول ج- اندیس د- رکورد

۴- کدام یک از گزینه های زیر برای انتخاب تمام فیلدهای یک جدول با نام tbl1 به کار می رود؟

الف- select tbl1 ب- select fields from tbl1

ج- select * from tbl1 د- select tbl1*

۵ - در دستور Where برای ادغام چند شرط، از چه دستوری استفاده می شود؟

الف-add

ب-or

ج-like

د- استفاده از چند شرط با نوشتن چند دستور where امکان پذیر است.

۶ - برای جستجوی افرادی در جدول که قسمتی از نام آنها محمد است چه عبارتی

مقابل دستور Where نوشته می شود؟

الف- " محمد" like ب- «محمد" like

ج- " محمد" like د- "محمد" like

۷ - برای درج یک رکورد در جدول tbl1 با فیلدهای f1 و f2 کدام یک از گزینه های

زیر به کار می رود؟

الف- select f1 , f2 from tbl1

ب- ("مقدار فیلد دوم" و "مقدار فیلد اول") values

ج- select into tbl1 (f1 , f2)

د- ("مقدار فیلد دوم" و "مقدار فیلد اول") values

ه- insert into tbl1 (f1 , f2)

و- ("مقدار فیلد دوم" و "مقدار فیلد اول") values

د- insert f1 , f2 from tbl1

("مقدار فیلد دوم" و "مقدار فیلد اول") values

۸- برای جستجوی یک رکورد در جدول tbl1 که مقدار دقیق فیلد آن را می دانید از

چه دستوری استفاده می کنید؟

الف- select * from tbl1

(... و "مقدار ۲" و "مقدار ۱") where f1 in

ب- select f1 from tbl1

(... و "مقدار ۲" و "مقدار ۱") where in

ج- search * from tbl1

(... و "مقدار ۲" و "مقدار ۱") where f1 in

د- select f1 from tbl1

(... و "مقدار ۲" و "مقدار ۱") alias in

۹- برای انتخاب ستون Name از جدول Student کدام عبارت به کار می رود؟

الف- name.student ب- student.name

ج- student*name د- name from student

۱۰- برای ادغام دو یا چند ستون مختلف از ۲ یا چند جدول و نشان دادن آنها در

یک ستون مشترک از چه دستوری استفاده می شود؟

الف- select ب- merge ج- join د- union

۱۱- برای حذف اجزای پایگاه داده از چه دستوری نمی توان استفاده کرد؟

الف- delete ب- insert ج- drop د- alter

۱۲- تابع sum و avg به ترتیب برای محاسبه چه عملی در پایگاه داده به کار می روند؟

الف- مجموع- مجموع ب- میانگین- مجموع

ج- مجموع- میانگین د- میانگین- میانگین

آزمون عملی

- ۱ - در Command Line یک پایگاه داده با عنوان Library ایجاد کنید.
- ۲ - برای موجودیت‌های کتاب و نویسنده، صفاتی که می‌توانند در محیط عملیاتی کتابخانه در نظر گرفته شوند را بررسی کنید و هر کدام از این دو موجودیت را به صورت جدول پیاده‌سازی کنید.
- ۳ - درون پایگاه داده Library که در تمرین ۱ ساخته‌اید دو جدول کتاب و نویسنده را با فیلدهای تعیین شده بسازید.
- ۴ - درون هر کدام از جدول‌های تمرین ۲ اطلاعات مربوط به چند رکورد را درج کنید.
- ۵ - تمام رکوردهای مربوط به جدول کتاب را نمایش دهید.
- ۶ - تمام رکوردهای مربوط به جدول نویسنده را که حرف اول نام نویسنده آن‌ها "H" است را نمایش دهید.
- ۷ - در جدول کتاب اسامی نویسندگان را نمایش دهید، طوری که اسامی تکراری حذف شوند.
- ۸ - در جدول کتاب اطلاعات مربوط به یک رکورد با عنوان "صفحات وب" را وارد کنید (سایر فیلدها را به دلخواه وارد نمایید)، سپس اطلاعات همان رکورد را تغییر دهید طوری که عنوان کتاب با مقدار جدید "طراحی صفحات وب مقدماتی" جایگزین شود.
- ۹ - در جدول نویسنده رکوردی که عنوان نویسنده آن با حرف "a" شروع می‌شود را حذف کنید.